

**UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS
ESCUELA DE INGENIERÍA INFORMÁTICA**

Proyecto de graduación

Para optar por el grado de Bachillerato en
Ingeniería en Sistemas de Información

**PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE VENTAS DE LA LIBRERÍA
Y BAZAR DIEKA UBICADA EN MORAVIA**

Diego Humberto Méndez Rodríguez

AUTOR

MAP Carlos H. Aguilar Mora

TUTOR

Lic. Daniel Álvarez Garro

LECTOR

San José, Costa Rica

Mayo, 2023

Contenido

DEDICATORIA	2
AGRADECIMIENTOS	3
CÓDIGO DE ÉTICA	9
Tablas	16
Figuras	18
RESUMEN EJECUTIVO	20
CAPÍTULO I: INTRODUCCIÓN	21
Planteamiento del problema	21
Objetivos	22
<i>Objetivo general</i>	22
<i>Objetivos específicos</i>	22
Justificación	22
<i>Viabilidad técnica</i>	23
<i>Viabilidad operativa</i>	23
<i>Viabilidad económica</i>	24
<i>Viabilidad legal</i>	25
<i>Proyecciones</i>	26
<i>Alcance funcional</i>	27
<i>Alcance metodológico</i>	29
<i>Alcance tecnológico</i>	29
CAPÍTULO II: MARCO REFERENCIAL	31
CAPÍTULO III: MARCO METODOLÓGICO	61
Enfoques de investigación	61
<i>Enfoque cualitativo</i>	61
<i>Enfoque de investigación seleccionado</i>	63
Método de investigación	64
<i>Investigación científica aplicada</i>	64
<i>Método de investigación seleccionado</i>	65
Fuentes de información	66
<i>Fuentes de información primaria</i>	66
<i>Fuentes de información secundaria</i>	67

	14
<i>Fuentes de información terciaria</i>	67
Unidades de análisis	67
<i>Variable conceptual</i>	68
<i>Variable operacional</i>	68
<i>Variable instrumental</i>	68
Instrumentos de recolección de datos	70
Proceso de recolección y análisis de datos	71
<i>Proceso de recolección de datos</i>	71
CAPÍTULO IV: ANÁLISIS DE RESULTADOS	73
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	77
Conclusiones	77
Recomendaciones	78
CAPÍTULO VI: PROPUESTA	79
Análisis detallado del software desarrollado	79
<i>Módulo de facturación</i>	79
<i>Módulo de compras</i>	80
<i>Módulo de cuentas por cobrar</i>	80
<i>Módulo de control de inventarios</i>	80
<i>Módulo de cuentas por pagar</i>	81
<i>Módulo de cierres de caja</i>	81
<i>Módulo de análisis de datos</i>	81
<i>Módulo de mantenimientos</i>	81
<i>Módulo de consultas</i>	81
<i>Módulo de reportes</i>	81
<i>Módulo de seguridad</i>	82
<i>Desarrollo del prototipo</i>	82
Análisis detallado del hardware requerido	83
Análisis detallado de los elementos relacionados con las telecomunicaciones	85
Descripción detallada de base de datos o almacenamiento de información	86
Descripción detallada del personal requerido para el uso del sistema	87
Diagrama de casos de uso	89
Casos de uso	90
Arquitectura del sistema	120

Arquitectura del software	121
Diseño de interfaces	122
Diseño de Base de datos	137
Diccionario de datos	138
Diagramas de Procesos	169
Diseño de salidas	177
<i>Diseño de salidas en pantalla</i>	177
<i>Diseño de salidas en formato PDF</i>	179
<i>Diseño de salidas en formato de Impresión</i>	181
Diagramas UML	182
<i>Diagrama de Clases</i>	182
<i>Diagramas de Actividades</i>	183
<i>Diagramas de Secuencias</i>	188
<i>Diagramas de Máquina de Estados</i>	193
Programación	194
<i>Entradas y Salidas</i>	194
<i>Procesos</i>	202
<i>Validaciones:</i>	207
Pruebas	210
APENDICES	220
Apéndice A. Guía de Entrevista Semiestructurada Aplicada.	220
Apéndice B. Guía de Observación Aplicada.	225

Tablas

Tabla 1. Versiones utilizadas para la elaboración del prototipo.	25
Tabla 2. Descripción de módulos del prototipo.	27
Tabla 3. Definición de unidades de análisis de la investigación.	69
Tabla 4. Requerimientos mínimos de software Visual Studio 2022.	82
Tabla 5. Requerimientos mínimos de hardware Visual Studio 2022	83
Tabla 6. Requerimientos mínimos de hardware de prototipo funcional.	84
Tabla 7. Características MSSQL Express Edition.	86
Tabla 8. Características MSSQL Standard Edition.	87
Tabla 9. Caso de uso de ingreso al sistema.	90
Tabla 10. Caso de uso de salida del sistema.	91
Tabla 11. Caso de uso de registro de usuarios.	92
Tabla 12. Caso de uso de consulta de usuarios.	93
Tabla 13. Caso de uso de modificación de usuarios.	94
Tabla 14. Caso de uso de eliminado de usuarios.	96
Tabla 15. Caso de uso de registro de clientes.	96
Tabla 16. Caso de uso de consulta de clientes.	98
Tabla 17. Caso de uso de modificación de cliente.	99
Tabla 18. Caso de uso de eliminación de cliente.	100
Tabla 19. Caso de uso de consulta de crédito de cliente.	101
Tabla 20. Caso de uso de registro de proveedores.	103
Tabla 21. Caso de uso de consulta de proveedores.	104
Tabla 22. Caso de uso de modificación de proveedores.	105
Tabla 23. Caso de uso de eliminación de proveedor.	107
Tabla 24. Caso de uso de consulta de crédito de proveedor.	108
Tabla 25. Caso de uso de registro de productos.	109
Tabla 26. Caso de uso de consulta de productos.	110
Tabla 27. Caso de uso de modificación de producto.	111
Tabla 28. Caso de uso de eliminación de producto.	113
Tabla 29. Caso de uso de registro de ventas.	113
Tabla 30. Caso de uso de registro de compras.	116
Tabla 31. Caso de uso de apertura de caja.	118
Tabla 32. Caso de uso de cierre de caja.	119
Tabla 33. Caso de uso de análisis de datos.	120
Tabla 34. Diccionario de datos, TUsers.	138
Tabla 35. Diccionario de datos, AspNetUsers.	139
Tabla 36. Diccionario de datos, AspNetUsersRoles.	140
Tabla 37. Diccionario de datos, AspNetRoles.	141
Tabla 38. Diccionario de datos, TProvincia.	142
Tabla 39. Diccionario de datos, TCanton.	142
Tabla 40. Diccionario de datos, TDistrito.	143
Tabla 41. Diccionario de datos, TClientes.	144
Tabla 42. Diccionario de datos, TDirecciónCliente.	145

Tabla 43. Diccionario de datos, TTelefonoCliente.	146
Tabla 44. Diccionario de datos, TCorreosClientes.	147
Tabla 45. Diccionario de datos, TCreditoClientes.	148
Tabla 46. Diccionario de datos, THistoricoPagosClientes.	149
Tabla 47. Diccionario de datos, TProveedor.	150
Tabla 48. Diccionario de datos, TDireccionProveedor.	151
Tabla 49. Diccionario de datos, TTelefonoProveedor.	152
Tabla 50. Diccionario de datos, TCorreosProveedor.	153
Tabla 51. Diccionario de datos, TCreditoProveedor.	154
Tabla 52. Diccionario de datos, THistoricoPagosProveedor.	155
Tabla 53. Diccionario de datos, TProducto.	157
Tabla 54. Diccionario de datos, TTipoProducto.	158
Tabla 55. Diccionario de datos, TRegistroCompras.	159
Tabla 56. Diccionario de datos, TDetallesCompras.	160
Tabla 57. Diccionario de datos, TTemporalCompras.	162
Tabla 58. Diccionario de datos, TRegistroDeVentas.	163
Tabla 59. Diccionario de datos, TDetallesVentas.	164
Tabla 60. Diccionario de datos, TTemporalVentas.	166
Tabla 61. Diccionario de datos, TRegistroCierre.	167
Tabla 62. Diccionario de datos, TRegistroApertura.	169
Tabla 63. Caso de prueba de inicio y cierre de sesión.	210
Tabla 64. Caso de prueba de CRUD de entidad de usuario.	210
Tabla 65. Caso de prueba de CRUD de entidad de cliente.	211
Tabla 66. Caso de prueba de CRUD de entidad de proveedor.	212
Tabla 67. Caso de prueba de CRUD de entidad de producto.	213
Tabla 68. Caso de prueba de consulta y ejecución de pagos a clientes y proveedores.	214
Tabla 69. Caso de prueba de cierre y apertura de caja.	215
Tabla 70. Caso de prueba de compras y ventas.	216
Tabla 71. Caso de prueba de análisis de datos.	217
Tabla 72, Apéndice B. Observaciones de hardware.	225
Tabla 73, Apéndice B. Observaciones de software.	226
Tabla 74, Apéndice B. Observaciones de empleados.	227
Tabla 75, Apéndice B. Observaciones de Ventas.	228
Tabla 76, Apéndice B. Observaciones de compras.	231
Tabla 77, Apéndice B. Observaciones de Inventario.	232
Tabla 78, Apéndice B. Observaciones de cierre de caja.	233

Figuras

Figura 1. Diagrama de casos de uso.	89
Figura 2. Diagrama de arquitectura del sistema.	121
Figura 3. Diagrama de arquitectura del software.	122
Figura 4. Pantalla de inicio de sesión.	123
Figura 5. Pantalla de recuperación de contraseña.	124
Figura 6. Pantalla de cambio de contraseña.	124
Figura 7. Pantalla principal.	125
Figura 8. Pantalla de lista de usuarios.	125
Figura 9. Pantalla de detalles de usuario.	126
Figura 10. Pantalla de actualizar usuario.	126
Figura 11. Pantalla de borrado de usuario.	127
Figura 12. Pantalla de lista de clientes.	127
Figura 13. Pantalla de registro de cliente	128
Figura 14. Pantalla de Detalles de cliente.	128
Figura 15. Pantalla de actualizar cliente.	129
Figura 16. Pantalla de borrado de cliente.	129
Figura 17. Pantalla de crédito de clientes.	130
Figura 18. Pantalla de listado de proveedores.	130
Figura 19. Pantalla de detalles de proveedor.	131
Figura 20. Pantalla de registro de proveedor.	131
Figura 21. Pantalla de actualizar proveedor.	132
Figura 22. Pantalla de borrado de proveedor.	132
Figura 23. Pantalla de crédito de proveedor.	133
Figura 24. Pantalla de registro de productos, compra.	133
Figura 25. Pantalla de procesamiento de compra.	134
Figura 26. Pantalla de listado de productos.	134
Figura 27. Pantalla de registro de producto.	135
Figura 28. Pantalla de detalle de producto.	135
Figura 29. Interfaz de Apertura de caja	136
Figura 30. Interfaz de Cierre de caja	136
Figura 31. Diagrama entidad - relación de base de datos	137
Figura 32. Diagrama de validación de rol	170
Figura 33. Diagrama de proceso de inicio de sesión	171
Figura 34. Diagrama de proceso ingreso de información.	172
Figura 35. Diagrama de proceso de actualización de información.	173
Figura 36. Diagrama de Proceso de borrado de información.	174
Figura 37. Diagrama de proceso ventas y compras.	175
Figura 38. Diagrama de Proceso cierre de caja.	176
Figura 39. Diseño de pantalla detalle de pago.	177
Figura 40. Diseño de pantalla reporte de compra.	178
Figura 41. Diseño de Salida de Reporte	179
Figura 42. Diseño de salida de pago de cliente.	180

Figura 43. Formato de ticket.	181
Figura 44. Diagrama de Clases.....	182
Figura 45. Diagrama de Actividades inicio de sesión.....	183
Figura 46. Diagrama de actividades registro de información.	184
Figura 47. Diagrama de Actividades actualizado de información.	185
Figura 48. Diagrama de actividades de borrado de información.	186
Figura 49. Diagrama de actividades de compra y venta.	187
Figura 50. Diagrama de secuencias de inicio de sesión.....	188
Figura 51. Diagrama de Secuencias de ventas.....	189
Figura 52. Diagrama de Secuencias de compras.	190
Figura 53. Diagrama de Secuencias de cierre de caja.....	191
Figura 54. Diagrama de Secuencias de apertura de caja.	192
Figura 55. Diagrama de máquina de estados.	193
Figura 56. Diseño de entrada de registro de clientes, primera parte.	194
Figura 57. Diseño de entrada de registro de clientes, segunda parte.....	195
Figura 58. Diseño de entrada de registro de clientes, tercera parte.	195
Figura 59. Diseño de entrada de registro de clientes, cuarta parte.	196
Figura 60. Clase de modelo de registro de clientes, primera parte.....	196
Figura 61. Clase de modelo de registro de clientes, segunda parte.	197
Figura 62. Clase de modelo de registro de cliente, tercera parte.....	197
Figura 63. Salida en pantalla de módulo de clientes, primera parte.	198
Figura 64. Salida en pantalla de módulo de clientes, segunda parte.	198
Figura 65. Salida en pantalla de módulo de clientes, tercera parte.	198
Figura 66. Salida en pantalla de módulo de clientes, cuarta parte.	200
Figura 67. Clase de salida en pantalla, primera parte.	200
Figura 68. Clase de salida en pantalla, segunda parte.....	201
Figura 69. Clase de generación de PDF.....	201
Figura 70. Formato de salida de Ticket.	202
Figura 71. Proceso de venta, primera parte.	202
Figura 72. Proceso de venta, segunda parte.....	203
Figura 73. Proceso de venta, tercera parte.	203
Figura 74. Proceso de venta, cuarta parte.	204
Figura 75. Proceso de venta, quinta parte.....	204
Figura 76. Proceso de venta, sexta parte.....	205
Figura 77. Proceso de cierre, primera parte.	205
Figura 78. Proceso de cierre, segunda parte.	206
Figura 79. Proceso de cierre, tercera parte.....	206
Figura 80. Validación de formatos.	207
Figura 81. Validación de existencia de datos.	207
Figura 82. Validación de rol.....	207
Figura 83. Validación de registros duplicados.....	207
Figura 84. Lista de áreas del proyecto.	208
Figura 85. Listado de "Data sets, primera parte.....	208
Figura 86. Listado de "Data Set"	209

RESUMEN EJECUTIVO

El siguiente proyecto de investigación fue orientado a la librería y bazar Dieka ubicada en el cantón de Moravia en la provincia de San José, trabajo el cual lleva por título “Prototipo funcional para la gestión de ventas de la librería y bazar Dieka ubicada en Moravia”.

La librería y bazar Dieka es un negocio de tamaño pequeño que lleva realizando sus operaciones más de 10 años en el mercado, ofreciendo servicios de venta de suministros de librería, bazar, regalos y centro de fotocopiado. Dentro de las distintas reuniones que se llevaron a cabo, con el fin de realizar la recopilación de requerimientos del prototipo se observaron varias debilidades en la línea de negocio que son muy vulnerables al error humano, tales como falta de control centralizado de información, falta de control computarizado de procesos o falta de reportes necesarios para la toma de decisiones.

Es por esto; por lo que el objetivo general del presente proyecto es elaborar un prototipo funcional que proporcione un soporte tecnológico para solventar dichas debilidades en el negocio utilizando una línea teórica de estudio que se basa en los procesos de ventas, compras, facturación, manejo de inventario y toma de decisiones con la ayuda de reportes generados mediante un sistema de información. En relación con lo mencionado, en el objetivo se desarrolló una investigación con un enfoque cualitativo que busco dar una solución a un problema planteado. Para llevar a cabo esta investigación se contó con una colaboración directa con la señora Roxana Barboza, dueña y administradora del negocio quien fue la encargada de suministrar todos los detalles de la operación.

Una vez finalizada la investigación se concluye que con la ayuda de un prototipo de software fue posible optimizar las tareas rutinarias del negocio y evitar errores humanos. Adicional a lo concluido, en esta investigación se recomienda al negocio implementar este prototipo para poder dar un soporte computarizado a sus operaciones de venta y así dar un mayor aprovechamiento del recurso humano y tiempo.

CAPÍTULO I: INTRODUCCIÓN

Planteamiento del problema

Librería y Bazar Dieka es un negocio mediano que se dedica a la venta de productos para librería, bazar, tienda de regalos y cuenta con una línea muy fuerte de servicios de impresión. Su ubicación posicionada cerca de varios colegios la hace ser uno de los negocios líderes de la zona. En los últimos años, su margen de ventas ha incrementado notoriamente; por lo tanto, el negocio ha enfrentado ciertos problemas en sus procesos de ventas que hoy se realizan por métodos muy manuales. Como facturas en papel, control de proveedores en libros y control de inventarios en cuadernos. A continuación, se presentan algunas de las problemáticas que estos procesos generan.

Actualmente, no se cuenta con un sistema centralizado que emita un reporte conciso para entregar a los clientes posterior a una compra, esto genera compradores insatisfechos, ya que no se emite una factura detallada de la compra realizada. De igual forma, existen muchos atrasos en las compras a crédito; por lo tanto, no se tiene certeza de los montos por cobrar. Asimismo, no existe un control centralizado para el inventario, por lo que, en muchas ocasiones, las existencias de un producto pueden llegar a cero y el comercio no logra percatarse a tiempo. Dentro del análisis de requerimientos realizados, se recopiló que en los últimos meses la empresa ha tenido una serie de quejas y, en su defecto, multas por parte de proveedores, pues, algunas veces, se incumplen fechas de pago por no tener un control claro de estas fechas. El comercio realiza el conteo de dinero a final de día; sin embargo, no se tiene otro insumo para poder corroborar estas cifras.

A partir de lo anterior, se concluye el planteamiento con la siguiente pregunta de investigación: ¿cómo un sistema de facturación, que permita digitalizar los procesos manuales de la empresa Dieka, le ayudaría a optimizar el proceso de ventas?

Objetivos

Objetivo general

- Desarrollar un prototipo funcional para la gestión de ventas de la Librería y Bazar ubicada en Moravia.

Objetivos específicos

- Analizar los requerimientos necesarios para la creación del prototipo funcional.
- Diseñar, según los requerimientos, el planteamiento estructural del prototipo que le permita a la librería la optimización de sus procesos de facturación.
- Programar el prototipo en todas sus capas para el cumplimiento de los requerimientos identificados.
- Realizar las pruebas funcionales necesarias del prototipo final.

Justificación

Librería y Bazar Dieka presenta actualmente una serie de problemas en sus procesos de ventas, compras, manejo de proveedores e inventarios. Algunos de estos inconvenientes pueden generar disconformidad en los clientes, porque no se emite una factura o recibo detallado al momento de hacer las compras en el comercio, además, puede haber proveedores molestos, ya que se han ocasionado atrasos en los pagos o fallos en los compromisos que se tenían con la librería; por otro lado, se acaban los productos en el inventario disponible y el comercio logra percatarse de este problema hasta que necesita dispensar alguno de estos artículos.

Los problemas mencionados en el párrafo anterior sirven como fundamento para realizar el presente trabajo de investigación, en donde, por medio de un prototipo funcional, se busca mitigar, o bien eliminar completamente las falencias que enfrenta actualmente el negocio Librería y Bazar Dieka ubicado en la Trinidad de Moravia.

Viabilidad técnica.

El proyecto es viable desde la perspectiva técnica, ya que requiere de pocos insumos para su utilización; únicamente, se requiere de un computador que cumpla con al menos las siguientes características:

- 8 GB de memoria.
- Sistema operativo Windows 10 o superior.
- Disco duro de estado sólido de al menos 250 GB.
- Procesador Intel CoreI5 o superior es recomendado.
- Dispositivos periféricos que se consideren necesarios (monitor, teclado, ratón...).
- Cajón de dinero e impresora de tiquetes (opcional, no es indispensable para el uso del sistema).

Actualmente, el comercio cuenta con un computador con las características mencionadas anteriormente; por lo tanto, no es necesario de la adquisición de *hardware* adicional. Se tiene planteado alojar el sistema en el mismo computador en el que se utilice, tomando los valores anteriores en cuenta reúne el suficiente procesamiento en relación con la carga de trabajo del negocio.

Desde el punto de vista de desarrollo, se utilizarán licencias para fines de desarrollo o estudio únicamente. Por lo que no es necesario contar con un entorno de desarrollo corporativo y con lo que un entorno de este tipo conlleva. Estas licencias de desarrollo pueden ser encontradas en el sitio oficial del proveedor que, para efectos de este prototipo, están disponibles para su descarga en el sitio oficial de Microsoft.

Viabilidad operativa.

El proyecto es viable desde una perspectiva operacional, ya que el prototipo por implementar efectuará las mismas operaciones que hoy en día se han realizado de una forma manual en la empresa. Con la implementación del prototipo, se espera optimizar, centralizar y automatizar los procesos, pero la forma de hacerlos seguirá siendo la misma con la mejora de hacerse por medio de un sistema que permitirá evitar errores humanos y facilitar la ejecución de estos procesos.

Los conocimientos que se requieren para la utilización de este prototipo es conocimiento básico de computación, tal como abrir una página web o como utilizar los

componentes periféricos del equipo (ratón, teclado, impresora...). Este conocimiento ya lo tienen los colaboradores de la empresa, pues una de las líneas de negocio que manejan, son los trabajos de impresión, por lo que ya de forma implícita conocen el manejo básico de un computador.

Se recomienda un entrenamiento de cómo utilizar el sistema posterior a su elaboración, el prototipo será desarrollado utilizando una interfaz amigable que le permita al usuario una curva de aprendizaje no mayor a unos días de entrenamiento. El sistema será utilizado por los empleados que atienden la librería. Sin embargo, este entrenamiento no va a ser cubierto dentro del alcance del presente trabajo de investigación, este punto se negociaría a futuro a lo interno con los encargados de la empresa en caso de requerirse.

Viabilidad económica.

El proyecto es viable en términos económicos, debido a que las licencias de desarrollo que se emplearán son de acceso comunitario para estudio o implementación de prototipos, siendo accesibles por cualquier persona en Internet. Estas licencias tienen todo lo necesario para la elaboración correcta del proyecto con una limitación de recursos que no estaría afectando para estos efectos. A razón de puesta a producción del prototipo (esta no está contemplada en el alcance del presente trabajo), se pueden utilizar los licenciamientos gratuitos tanto de IIS Express, y la versión de SQL Standard Edition cuyo precio en el mercado ronda los \$899usd, ya que, al ser una empresa pequeña, no se requiere de gran procesamiento ni características avanzadas. Adicional, para el módulo de análisis de datos se requiere adquirir una licencia de Power BI que ronda los \$10usd por mes. El negocio ya posee una cuenta de Microsoft corporativa cuya licencia tiene un precio de \$36usd por mes, por lo que no es requerido adquirir esta licencia.

A continuación, se observa la siguiente tabla con las licencias que se van a utilizar para la implementación del prototipo:

Tabla 1. *Versiones utilizadas para la elaboración del prototipo.*

Aplicación	Tipo de licencia	Versión
IDE Visual Studio 2022	Community Edition	2022
Web Server Microsoft IIS	Express Edition	10.0
Database Engine Microsoft SQL Server	Community Edition	2019
Power BI	Pro	2023

Fuente: Elaboración propia.

En la tabla anterior se pueden observar las licencias que serán utilizadas durante la elaboración del proyecto, estas no tienen costo alguno.

A razón de compra de equipo, el costo en el mercado de adquirir hardware con las características especificadas en la viabilidad técnica ronda los 450 000,00 colones, aproximadamente. Sin embargo, la empresa ya cuenta con un equipo que reúne las características mínimas necesarias para correr el sistema por lo tanto no es requerido incurrir en un gasto de adquisición de equipo físico.

A razón de cobro del recurso humano, el costo por día de elaborar este prototipo entra un aproximado de 14 381,27 colones costarricenses por un día laborado de 8 horas, esto corresponde a 1797,65 colones costarricenses por hora laborada, esta información corresponde al salario mínimo de a un programador de computación sin título, esto según el Ministerio de Trabajo de Costa Rica. En una estimación basada en el costo por hora laborada se estima un costo total del prototipo en 143 812,00 colones, aproximadamente, en un promedio de 80 horas trabajadas. Sin embargo, es importante acotar que estos costos de recurso humano no entrarían en vigor en la elaboración del prototipo, pues la elaboración del prototipo se trata de un proyecto final universitario.

Viabilidad legal.

El proyecto será realizado dentro del marco legal estipulado para el sector privado, en donde el negocio tiene apertura de operar libremente, mientras esté dentro del marco legal para el sector privado en la jurisdicción costarricense. Con respecto a los reportes de facturación que se emitirán en el sistema, estos serán diseñados tomando en cuenta lo que

estipule el Régimen Tributario Simplificado, que es el que corresponde a la empresa a la cual está dirigido este proyecto de investigación.

El sistema cumplirá lo estipulado en las leyes N° 8968, Ley de Protección de la Persona frente al tratamiento de sus datos personales, y la Ley N° 8148, sobre delitos informáticos. Esto se logra debido a que el prototipo contará con todas las previsiones correspondientes a seguridad tanto a nivel de aplicación como en términos de bases de datos, utilizando técnicas como encriptación, uso de usuario y contraseña para ingreso al aplicativo y, de igual forma, los datos no estarían públicos a terceros ni en Internet. En resumen, las siguientes leyes rigen la creación del presente prototipo.

- Ley N° 8148, Adición de los artículos 196 BIS, 217 BIS y 229 BIS al Código Penal.
- Ley N° 4573, para reprimir y sancionar los delitos informáticos de la Asamblea Legislativa de la República de Costa Rica del año 2001.
- Ley N° 6683, de Derechos de Autor por parte de la Asamblea Legislativa de la República de Costa Rica del año 1982.
- Ley N° 8968, sobre la protección de la persona frente al tratamiento de sus datos personales.

Proyecciones

Con el proyecto de investigación, se pretende optimizar las operaciones de facturación de la empresa en las cuales actualmente se realizan de forma manual, al transformar estas operaciones y llevarlas a cabo a través de un sistema se asegura mayor robustez en el proceso en donde exista menor posibilidad de error humano, pérdida de información y agilidad al momento de entregar reportes o evaluar indicadores. De igual forma, con la utilización de un sistema, se espera tener un mayor control de las operaciones que realizan los empleados que operan en la empresa.

La empresa recibirá un beneficio directo con el prototipo, ya que, al contar con un sistema de facturación, los clientes estarán más conformes con el reporte que reciben, con esto, pueden recibir la factura por medio de correo electrónico en donde pueden visualizar con detalle los ítems adquiridos en el negocio. De igual forma, el sistema lleva un mayor seguimiento de los pagos a proveedores, por lo que la empresa estaría disminuyendo en gran medida posibles multas por atraso en pagos a proveedores.

Otro beneficio importante que va a contar el negocio posterior a la implementación del prototipo es un mayor control sobre el inventario, en donde se asegure que, si un ítem se está quedando sin existencias, este pueda ser encargado a tiempo a los respectivos proveedores.

Alcance funcional.

El prototipo está diseñado para cubrir las necesidades detectadas en el análisis de requerimientos en relación con sus operaciones de facturación; en la siguiente tabla, se definen los diferentes módulos con los que contará el prototipo en este proyecto de investigación:

Tabla 2. *Descripción de módulos del prototipo.*

Nombre del módulo	Descripción del módulo
Generar factura	Este módulo se encarga de emitir las facturas por las ventas de los clientes, dicho módulo cargará el inventario previamente definido donde se podrán escoger los ítems vendidos. Se mostrará el valor de cada ítem, y, al finalizar, la venta mostrará información totalizada de descuentos, impuestos, total bruto y neto de la venta. Este módulo también se encargará de emitir el tiquete electrónico relacionado con la facturación.
Generar compra	Este módulo se encarga del procesamiento de compras para la adquisición de productos en el inventario. Dicho modulo tiene que ser capaz de cargar la información de los proveedores y los productos respectivos a realizar la compra. Se mostrará el valor de cada ítem, y, al finalizar, la compra mostrará información totalizada de descuentos, impuestos, total bruto y neto de la compra.
Cuentas por cobrar	Este módulo se encarga de llevar el control de las cuentas por cobrar, se registran abonos sobre los créditos, se estipulan las fechas de pago, se calculan los montos de morosidad en caso de que apliquen, y envía notificaciones al cliente y al encargado del negocio sobre fechas de pago cerca a vencerse. De igual forma, el módulo está ligado directamente con

	el módulo de facturación para registrar los abonos realizados en la contabilidad del negocio.
Control de inventario	Este módulo se encarga de mantener un control de los artículos, es decir, la cantidad de cada producto por la línea de producto, en el inventario, disminuyendo ante ventas y aumentando ante compras a proveedores. Cuando un artículo está llegando a su mínimo, se les envía un correo a los encargados notificando la cantidad de artículos disponibles. Se utilizarán puntos de reabastecimiento.
Cuentas por pagar	Este módulo se encarga de calendarizar los pagos a proveedores y mantener un control detallado de las cuentas por pagar que tiene el negocio con los proveedores. Cuando un pago está próximo a efectuarse, se avisa con antelación mediante correo electrónico.
Cierre de caja	Este módulo puede generar de forma automática o manual los cierres de caja, cuando el usuario ingresa, este se crea, haciendo responsable por el dinero que ingresa y sale de la caja. De esta forma, el sistema se asegura que todas las facturas pertenezcan a un mismo cierre.
Análisis de datos	Este módulo permite por medio de gráficas sencillas de interpretar determinar tendencias en la venta y compra de productos que ofrece el negocio.
Mantenimientos	Este módulo se encargará de realizar el borrado, inserción, modificación, actualización de datos de proveedores, clientes, artículos, líneas de artículos, entre otros.
Consultas	Este módulo se encargará de generar información proporcionada de las diferentes tablas.
Reportes	Este módulo se encargará de generar información proporcionada de las diferentes tablas y procesos, pero con un formato específico, según lo solicite el usuario. Podrá ser impreso o por pantalla.
Seguridad	Este módulo se encargará de realizar la autenticación de contraseñas y definición de perfiles.

Fuente: Elaboración propia.

Alcance metodológico.

El método utilizado en la elaboración del prototipo es el denominado “Ciclo de vida”, según el Laboratorio Nacional de Calidad del Software del Instituto de Normas Técnicas de Costa Rica [INTECO] (2009), se enumeran las siguientes etapas:

1. Recolección de requerimientos.
2. Especificaciones.
3. Análisis.
4. Diseño.
5. Implementación.
6. Pruebas.
7. Mantenimiento

Dentro de la elaboración del prototipo, aplicarán los primeros seis pasos mencionados en el listado anterior; sin embargo, no estaría teniendo efecto el paso siete correspondiente al mantenimiento del prototipo, ya que es responsabilidad del usuario final dar el mantenimiento correspondiente al sistema.

La metodología de ciclo de vida es un proceso integral de desarrollo de *software* que va desde las reuniones con el cliente para conocer las diferentes problemáticas que enfrenta el negocio, hasta la última etapa de puesta a producción y mantenimiento del producto realizado. De todos los métodos de programación es el que asegura que se cumplan en calidad y forma todos los requerimientos establecidos por el cliente; sin embargo, en comparación con otras metodologías ágiles, como *scrum*, suele ser una metodología más laboriosa y lenta de llevar a cabo por la cantidad de documentación y pruebas que esta conlleva, a pesar de esta desventaja en comparación con otras metodologías ágiles se escogió este método, ya que permite un detalle muchísimo mayor de cada uno de los componentes del sistema, y su interacción con las diferentes etapas del ciclo de vida.

Alcance tecnológico.

El prototipo será desarrollado como una aplicación web, no obstante, este, por cuestiones de presupuesto del cliente. será alojado en el mismo equipo donde se ejecutará el *software*, como es una empresa pequeña y no tiene una gran cantidad de personal no es requerido realizar un *hosting* del aplicativo. Se utilizarán las siguientes tecnologías:

- ASP.NET MVC en C#

- Entity Framework
- Microsoft SQL Server
- Power BI Pro Edition

Una de las principales ventajas de utilizar estas tecnologías es debido a que Microsoft ofrece una interfaz muy amigable para el desarrollador, existe gran cantidad de documentación por parte del proveedor, y es un lenguaje bastante optimizado. En caso de que se requiera realizar el *hosting* del sitio, este se tiene que gestionar como un proyecto aparte y no está incluido dentro de los alcances de esta investigación.

CAPÍTULO II: MARCO REFERENCIAL

Como en todos los campos de la vida cotidiana, la tecnología está otorgando un valor importante a las operaciones cotidianas de los mercados, dentro del campo de ventas es de suma importancia contar con un sistema centralizado que permita a las empresas dedicadas al mercadeo y ventas un control exacto de las cantidades de dinero que entran o salen de la compañía, así como la cantidad de artículos que tienen almacenados en los inventarios. A estos sistemas, comúnmente, se les conoce, de igual manera, como Puntos de Venta (POV). En la siguiente cita, Aguilar y Anaya (2015) indican, en términos generales, lo que se conoce como punto de venta:

Es un sistema compuesto por *software* y *hardware*, creado especialmente para agilizar los procesos relacionados con ventas, compras, inventarios, créditos, cobranza, cotizaciones, lotes, y atenciones al cliente, entre otras cosas. Un punto de venta es un sistema informático (POS) o electrónico micro computarizado (ECR) que gestiona el proceso de venta mediante una interfaz accesible para los vendedores o compradores. (Aguilar y Anaya, 2015, p. 9).

En este sentido, se comprende que un sistema de gestión de ventas es un entorno integral compuesto por componentes físicos (*hardware*) y lógicos (*software*), en donde se desarrolla un sistema para facilitar las operaciones de venta, tales como facturación, compras, control de inventarios y la emisión de reportes tanto para el cliente como para el negocio. En el caso de la Librería y Bazar Dieka, en donde se maneja un gran número de operaciones de forma manual, este tipo de sistemas viene a solventar una gama de inconvenientes que puede ser ocasionados por el fallo humano y que, al otorgarle esta responsabilidad a un *software*, pueden evitarse o disminuirse notablemente los errores de proceso. A manera de ejemplo y para el caso práctico de la librería, esta lleva un registro de facturas en papel al momento de la venta de un artículo, esta forma de realizar el proceso puede presentar fallos, tales como pérdidas de documentos o colocación de precios incorrectos de una factura, en contraparte, desde un sistema de punto de venta, las facturas quedan guardadas en una base de datos centralizada (que puede ser incluso respaldada), en donde la información que se coloque en cada factura fue definida previamente por el administrador.

Al respecto, la presente investigación plantea el diseño y desarrollo de un prototipo de la naturaleza de un punto de venta, en donde se busca mitigar una serie de problemáticas para una empresa. Según Salazar (2012), un prototipo es el “primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otros iguales, o molde original con el que se fabrica” (p. 2).

De tal modo que un prototipo es la implementación o invento de algún objeto sea tangible o no, en los cuales esta primera versión está sujeta a cambios, actualizaciones, mejoras o modificaciones en la versión inicial. Este tipo de implementaciones se utiliza en gran medida para realizar desarrollos rápidos en las que se necesita probar de manera ágil una solución a algún problema. Para el caso práctico de esta investigación, se busca elaborar un prototipo inicial que permita dar por solución a una serie de problemas identificados en los procesos gestionados por la Librería y Bazar Dieka. Como se mencionó anteriormente, su naturaleza es un primer ejemplar que abre la posibilidad a futuro de sufrir cambios o alteraciones del sistema a razón de necesidades del negocio.

Dentro del proceso de la elaboración de un prototipo, es importante tomar algunas consideraciones preliminares, al momento de implementar un prototipo que se indican a continuación:

Problemas no estructurados, novedosos y complejos, de información personalizada del usuario, ya que sus salidas no son predecibles y definidas.

Problemas de ambiente Inestable, el profesional también debe evaluar el contexto del sistema.

Experiencia en diseños similares.

No se conocen los requerimientos, la naturaleza del sistema es tal que existe poca información con respecto a las características que debe tener el nuevo sistema para satisfacer las necesidades del usuario.

Los requerimientos deben evaluarse, se conocen los requerimientos aparentes de información, pero es necesario verificarlos y evaluarlos.

Costos altos, donde la inversión involucra gran cantidad de recursos financieros y humanos.

Altos riesgo, la evaluación inexacta de los requerimientos o el desarrollo incorrecto ponen en peligro a la organización.

El usuario no está dispuesto examinar modelos en papel, o no sabe lo que quiere, pero lo reconocerá cuando lo vea.

Tecnologías Nuevas: la falta de experiencia en el uso de dichas tecnologías, junto con el deseo de instalar nueva tecnología hace que sea propicio el uso del prototipo (Salazar, 2012, p. 2).

Dicho de otro modo, cuando surge la necesidad de implementar un prototipo con el fin de optimizar un proceso, cerrar alguna brecha de seguridad o bien disminuir la posibilidad de error humano, es indispensable hacer un análisis exhaustivo preliminar antes de dar comienzo a la elaboración del prototipo, esto ya que cabe la posibilidad de que, sin este análisis previo al generarse el prototipo, se ejecuten las pruebas respectivas y se determine que, más bien, resulta contraproducente para el buen desempeño del negocio la implementación del prototipo. Dentro del marco de esta investigación, se realizó un análisis exhaustivo, tomando las consideraciones mencionadas anteriormente, donde se recapitularon los procesos actuales que ejecuta la Librería y Bazar Dieka, este punto resulta de vital importancia, pues el prototipo que se pretende generar debe tener una similitud sistemática a los procesos actuales, pero llevados a un sistema de información. Esto pretende disminuir en gran medida la resistencia al cambio por parte del usuario y, por otro lado, no se incurre en una gran modificación de los procesos actuales de negocio. Por su parte, al realizarse un estudio de otros sistemas similares en el mercado, se acopla el prototipo con todas las funcionalidades necesarias para el buen aprovechamiento del negocio.

Una de las áreas del negocio que se estará tomando en cuenta para la elaboración del prototipo es la gestión correcta de productos, manejo de inventario o tratamiento de inexistencias. El producto es la base de todo comercio en donde se espera recibir una retribución monetaria, por la creación o distribución de ese producto. De acuerdo con lo mencionado por el profesor Santesmases (2009), un producto es “cualquier bien material, servicio o idea que posea un valor para el consumidor y sea susceptible de satisfacer una necesidad” (p. 7). Consecuentemente, el producto es aquel bien material o servicio, que el consumidor desea obtener y tiene que pagar un costo para poder recibirlo. Uno de los primeros pasos en la creación de una idea de negocio, es decir, delimitar un público meta y establecer el tipo de producto, marcas o naturaleza de este despertan interés en este público

meta. Por lo tanto, la correcta colocación del producto en el consumidor adecuado representa la clave de un negocio exitoso.

Los productos tienen los siguientes atributos con los que pueden ser clasificados:

1. Núcleo: se refiere a las propiedades físicas, químicas y técnicas del producto.
2. Calidad: valoración de los elementos que componen el núcleo, de acuerdo con unos criterios que son comparativos con la competencia.
3. Precio: valor último de adquisición.
4. Envase: elemento de protección del que está dotado el producto y que, junto al diseño, aporta un gran componente de imagen.
5. Diseño, forma y tamaño: permiten la identificación del producto o la empresa y, generalmente, configura la propia personalidad de este.
6. Marca, nombres y expresiones gráficas: igualmente, facilitan la identificación del producto y permiten su recuerdo asociado a uno u otro atributo.
7. Servicio: conjunto de valores añadidos a un producto que nos permite poder marcar las diferencias respecto a los demás.
8. Imagen del producto: opinión que se crea en la mente del consumidor según la información recibida, directa o indirectamente, sobre el producto.
9. Imagen de la empresa: opinión en la memoria del mercado que interviene positiva o negativamente en los criterios y actitudes del consumidor hacia los productos. Una buena imagen de empresa avala, en principio, a los productos de nueva creación; así como una buena imagen de marca consolida a la empresa y al resto de los productos de esta (Pérez y Pérez, 2006, p. 8).

Comprender cada uno de estos atributos es de vital importancia para tener los resultados esperados en la generación de utilidades de la empresa, ya que cada atributo va a ser determinante para cada tipo de población o consumidor meta. Por ejemplo, cierto foco de población prefiere la adquisición de ciertas marcas que otras, o se ve inclinado a adquirir productos de cierto precio, confiabilidad o calidad. El no contar con un sistema de información que almacena todos estos atributos para cada elemento, es muy complicado recordar o conocer en su totalidad los productos que se venden en el comercio, dentro de un sistema de punto de ventas se pueden recapitular todos los productos que se tiene en existencias y guardar sus atributos tales como precio, proveedor, descripción física o química,

marca, naturaleza del producto, en caso de ser perecederos, información elemental como fechas de vencimiento para un respectivo lote de productos. En esta perspectiva, se busca implementar dentro del prototipo desarrollado para Librería y Bazar Dieka un módulo en donde se pueda recapitular toda esta información y almacenarla en un sistema centralizado, de esta forma se pueden construir reportes o, incluso, hacer gráficos de análisis que permita al comercio ver qué producto es el más vendido, o qué marca es la que prefieren los consumidores; estos indicadores pueden resultar beneficiosos para buscar mejorar relaciones comerciales con determinados proveedor o, en su defecto, dejar de adquirir cierta marca o producto, ya que no está generando las utilidades esperadas.

El producto es únicamente la base de un buen negocio, en este punto, se tienen que contemplar aspectos como la calidad o los intereses de los clientes; sin embargo, es necesario una correcta colocación de estos productos en el mercado, es por este motivo que otras de las variables que son contempladas dentro de la elaboración del prototipo es la gestión de ventas para la colocación de productos en los clientes, en palabras de Vega Sánchez (s.f.), la gestión de ventas “es la acción de vender. Es traspasar a otro la propiedad por el precio convenido” (p. 5). En otras palabras, la naturaleza del proceso de ventas es el poder generar utilidades por medio del intercambio de productos o servicios, asignándole un precio respectivo para la venta. El proceso de ventas es uno de los procesos más naturales y fundamentales en una empresa así sea materializado en forma de producto, o bien brindando algún servicio con el que se espera obtener una retribución monetaria a cambio. Vega Sánchez (s.f.) adiciona lo siguiente en cuanto a la importancia del proceso de ventas:

Hay un refrán en los negocios que afirma que no sucede nada sino hasta que se efectúa una venta. Sin ventas no se necesitan contadores, trabajadores de producción, ni siquiera el presidente de la compañía. Las ventas representan el combustible que hace que funcionen los motores empresariales (p. 7).

Tal como lo menciona Vega Sánchez (s.f.), el proceso de ventas representa el pilar fundamental de toda función empresarial; por lo tanto, es importante y resulta indispensable una correcta ejecución de la gestión de ventas en donde se asegure la agilidad, la precisión y la seguridad en los procesos de ventas, asimismo, que se tenga los insumos necesarios para poder realizar de forma continua la estrategia de ventas.

Dentro de este orden de ideas, se establece como un módulo fundamental para el prototipo una correcta gestión de ventas, en donde, en lo recopilado en los procesos del negocio, se observan escenarios como falta de control de un registro sistematizado de ventas, o bien facturas emitidas en papel, en donde estas son archivadas y, al perderse o dañarse, estos archivos físicos se pueden perder información importante que puede ser irrecuperable. Es por esto que, dentro de la solución, se propone enviar toda esta información a un *software* de punto de venta, esta va a ser recibida, procesada y, luego, almacenada en un lugar en donde se pueden definir respaldos en caso de un siniestro con la información, a su vez, guardando un histórico de ventas que puede ser consultado en cualquier momento, por otro lado, se hace hincapié en la veracidad y estética del documento y factura emitida al usuario, en donde, por medio de un sistema de gestión de venta, se puede desarrollar una factura con un diseño más amigable a la vista del usuario, alimentada con información previamente configurada y validada por el administrador. Según Serrano (2010), el proceso de ventas se puede dividir en las siguientes fases:

1. Búsqueda de clientela: Prospección de clientes, en esta fase se plantean los clientes potenciales que tienen la capacidad y la necesidad de adquirir.
2. Preparación de la venta: Estudio de las necesidades del cliente, una vez realizado el listado de todos los clientes potenciales es necesario realizar el estudio de cuáles son las necesidades actuales de estos clientes, y elaborar una propuesta creativa en la que se pueda suplir estas necesidades.
3. Contacto: Realizar el primer contacto con el cliente, realizar una presentación formal de la empresa y una demostración de los productos a ofrecer.
4. Desarrollo: Entender las necesidades del cliente, informar, presentar y demostrar. En esta fase se tiene que realizar un estudio personalidad acorde a las necesidades del cliente para poder otorgar una oferta final.
5. Cierre: Obtener el objetivo propuesto en la venta, y terminar la reunión con aportes del cliente.
6. Análisis: Efectuar una revisión detallada de la visita con el cliente, del proceso y del resultado obtenido. Esta fase permite una mejora continua en la estrategia de negocio.

7. Servicio post venta: Brindar ayudas y soporte para garantizar la satisfacción del cliente, es importante este punto ya que el proceso de estrategia de ventas no termina una vez entregado el producto. Si no, que es necesario dar un seguimiento del cliente para mantener la satisfacción de este (p. 16).

Por consiguiente, para poder construir y elaborar una correcta gestión de ventas, hay que tener en claro los métodos de acción en cada una de estas fases, estableciendo el enfoque del público meta como punto inicial hasta otorgar el servicio o producto en calidad y forma tomando en cuenta lo que el cliente necesita y de la forma en la que lo requiere. Con el prototipo a realizar, se puede fortalecer cada una de estas fases gracias a las facilidades que otorga un sistema de punto de venta, como lo es un correcto directorio de clientes para la fase de contacto, un módulo de análisis que permita conocer las necesidades del cliente en relación con los productos o servicios obtenidos en un determinado tiempo, de esta forma, dando soporte a las fases de desarrollo y análisis, así como un correcto procedimiento al momento de la venta en términos de reporte otorgado al cliente y veracidad de la información, que pueda fortalecer las etapas de cierre y servicio post venta.

En acompañamiento con el objetivo de entender cada una de las fases del proceso de ventas mencionadas en el párrafo anterior, el manual de atención al cliente y del proceso de ventas publicado por Wellness & Sport Consulting (2021), existen 10 aspectos fundamentales que se deben considerar para tener el resultado esperado en un proceso de venta:

1. Cumplir con las promesas: Realizar promesas que no se tiene la certeza de poder cumplir es un hábito frecuentemente utilizado para evadir un problema. Sin embargo, en una venta el no cumplir con una promesa puede traer resultados negativos en la relación del cliente y la empresa.
2. Respetar el tiempo del cliente: El tiempo es el valor máspreciado por el cliente y uno de los recursos más escasos. Aquellas empresas que quieran dar un valor real al producto que se ofrece, debe saber gestionar el tiempo de los clientes. Por lo tanto, un servicio optimizado, tendrá mayor satisfacción en el cliente.
3. Tener obsesión por los detalles: El concepto de calidad se basa en la necesidad de sorprender al cliente. Este elemento de sorpresa se consigue con pequeños detalles que den un componente emocional al servicio brindado.

4. Ser siempre cortés: El trato que se le da el cliente aumenta las probabilidades que ese cliente vuelva a adquirir un producto o un servicio en la compañía. Por lo tanto, velar siempre por el trato correcto con el cliente asegura la continuidad del servicio otorgado.
5. Mantener siempre el rol profesional: Ser un buen profesional significa hacer lo que se tiene que hacer, de la manera como se tiene que hacer, siempre y con todas las personas. Se tiene que velar por dar el mejor servicio al cliente todos los días. Mantener la calidad de un servicio profesional es indispensable para una buena reputación de la compañía.
6. Dar seguridad al cliente: Se puede conseguir que el cliente se sienta seguro si el vendedor y todos los recursos de la empresa se ganan la confianza del cliente. Esto se logra evitando engaños al cliente, realizar promesas que no se cumplen. Es indispensable asegurar que las transacciones que vaya a realizar el cliente sean seguras y confiables.
7. Respetar la confidencialidad: Este punto está estrechamente ligado al punto anterior. El cliente debe tener la certeza de lo que todo lo que se hable con el ejecutivo de la compañía quedará en la intimidad y que no se divulgarán los datos aquí expuestos a terceras personas.
8. Ser accesibles: Ser accesibles tanto físicamente como en términos de actitud de todos los ejecutivos que laboran en la compañía. En donde el cliente fácilmente pueda realizar sus consultas abiertamente, sin incurrir en procesos muy complejos para poder ingresar a los servicios de la compañía.
9. Comunicación frecuente y comprensible: El cliente necesita recibir información continuamente. Uno de los principales motivos por los que un cliente abandona la compañía es por falta de comunicación y no sentir un acompañamiento real por parte del ejecutivo.
10. Reacción frente al error: En todas las organizaciones se producen errores, por lo que la presencia de estos es un acto inevitable. Sin embargo, en esta área el cliente toma valor de cómo la compañía enfrenta y soluciona estos errores (p. 4).

Se concluye que, cumpliendo en forma estos 10 aspectos, aumenta la posibilidad de obtener un buen resultado al momento de seguir cada una de las fases del proceso de ventas,

algunos de estos aspectos hacen referencia a un término conductual del vendedor con el cliente; sin embargo, en otros aspectos, tales como la seguridad, la comunicación, la confidencialidad y el manejo al fallo, se pueden mejorar con la implementación de un sistema centralizado de gestión de ventas, en donde, al disminuir la cantidad de operaciones manuales, el comercio pueda invertir tiempo y recursos, poniendo atención en los detalles hacia el cliente para mantenerlo a gusto con el servicio recibido y, de igual manera, realizar búsquedas de nueva clientela y nuevos productos.

Una vez entendida la importancia de una correcta gestión de ventas, cuando un negocio ya se encuentra posicionado en el mercado, es relevante no perder de vista que gran parte de ese éxito se debe a la relación comercial con los proveedores que ya sea venden materia prima para la manufactura de un producto final, o bien venden el objeto manufacturado para su redistribución. Es por esto por lo que otra de las claves para un negocio exitoso es tener una correcta gestión de compras y manejo de proveedores. Según Sangri (2014), “el término comprar se define de la siguiente forma, del latín: *comprare*-adquirir. Y significa conseguir y obtener, acción de adquirir y hacer propio a lo que a nadie le pertenece” (p. 13).

Como se puede observar en lo expuesto, la acción de comprar es conseguir algún producto o servicio por medio del intercambio de dinero o bien de un valor de interés de la parte proveedora del bien. Un correcto proceso de compras impacta directamente, ya sea de forma positiva o negativa, en el proceso de ventas de una compañía, pues la calidad, el precio y la eficacia de los productos ofrecidos al consumidor dependen de los proveedores que suplen estos productos, o bien distribuyen la materia prima necesaria para la fabricación de este. Sangri (2014) adiciona lo siguiente en referencia a la importancia de la correcta gestión de compras:

Las compras bien planeadas deben redituarse a la empresa ahorros en efectivo, en su liquidez, y en la fluidez del capital. Bajo un sistema organizado, las compras le representan a la empresa una buena administración, negociando plazos de pago, descuentos, oportunidad de uso y otros beneficios. Es vital para la adquisición de las materias primas y los recursos necesarios, involucrando los elementos que determinan la coordinación de las áreas de la empresa que requieren de materiales y/o insumos para su eficiente funcionamiento (p. 13).

Dentro de este orden de ideas, es importante adicionar que en este proceso de la compañía es indispensable la búsqueda continua de la mejor oferta y demanda de productos y servicios en el mercado, realizar un estudio detallado de las diferentes opciones que ofrecen los distintos proveedores y escoger la opción que más conviene según las actividades del negocio. Para cuidar las relaciones con los proveedores es de vital importancia que el comercio mantenga un histórico de pagos intacto esto le ofrece al comercio la posibilidad de apertura de nuevos créditos y distintas aplicaciones de descuentos. En relación con los procesos que actualmente maneja la librería, el proceso de control de compras y proveedores se maneja en libros y archivos físicos, esto representa un riesgo de errores humanos en el registro de la información o pérdida de esta, es por esto que, en función del prototipo a desarrollar para la librería, se propone un módulo de compras, con el fin de optimizar el proceso de control de compras que actualmente maneja la librería, en donde, trasladando la operatividad al *software*, se pueden mejorar procesos de registro de clientes en forma de un directorio, en el que se tenga un sitio centralizado que contenga a todos los proveedores del comercio, por otro lado, la posibilidad de un registro de pagos a proveedores de una forma más ordenada, en donde se puedan visualizar claramente las distintas fechas de pago en caso de contar con líneas crediticias para un determinado proveedor.

Utilizando el aprovechamiento que puede traer el prototipo y acorde al análisis expuesto por Sangri (2014), una correcta gestión del proceso de compras puede buscar los siguientes objetivos:

- Reducción de costos: Obtención de utilidades y a la vez mejorar el trinomio costo-beneficio-utilidad.
- Compra al mínimo precio: Un precio adecuado por la adquisición de los artículos, sin perder el foco en los factores de calidad, volumen, y pronta utilización de lo adquirido.
- Servicio: Optimizar los tiempos y las coordinaciones con los departamentos a los que les da el servicio, para con ello cumplir con los requerimientos solicitados.
- Responsabilidad del área de compras: La búsqueda de mejores precios, proveedores, pedidos, inspección de lo adquirido y reportes para la compañía tales como Informes de compras, inventarios, contratos o almacenaje. Son de las principales responsabilidades que tiene que velar el departamento de compras.

- Control de convenios: En este objetivo al área de compras le interesan todos los convenios que fueron, son y serán necesarios para el financiamiento que otorgue el proveedor. Estos convenios pueden considerar aspectos como, tiempo de pago, precios accesibles, planes de lealtad.
- Control de tratados comerciales con los proveedores: En lo referente a buenas relaciones, ética, mantener una política de varios proveedores, aseguramiento del prestigio e imagen de la empresa (p. 14).

En conclusión, cada uno de estos objetivos buscan un mismo fin, el cual es lograr una adquisición inteligente de los productos, y para esto la mejor forma de realizar un análisis de datos reales es por medio de un sistema centralizado que haga esto posible, de igual forma, haciendo hincapié en que en el objetivo de control de convenios y tratados comerciales es de suma importancia mantener las relaciones crediticias intactas. Otro aspecto importante dentro de una correcta gestión de compras es poder identificar dónde se encuentran esos puntos relevantes al momento de la toma de decisiones en la adquisición de nuevos productos o servicios, cuando se habla de adquisición de inventario, Sangri (2014) hace referencia a 3 pilares fundamentales en los cuales se recomienda basar las operaciones de compra en una compañía:

1. Calidad: Base para que los productos que fábrica la empresa o revende, sean los que el consumidor prefiere, bajo las condiciones que el cliente espera.
2. Cantidad: Factor de suma importancia que contempla atributos tales como cantidad de almacenaje en los almacenes, cantidad de acuerdo con fechas de caducidad, y mientras mayor sea la cantidad de productos adquiridos se debe proporcionar un mejor precio.
3. Precio: Relación a la calidad, cantidad, fechas de entrega y cobro (p. 17).

Se plantea, entonces, que el saber identificar estos tres pilares conducen al negocio a tomar mejores decisiones con respecto a la compra de productos y servicios para llevar a cabo sus operaciones, estos tres pilares fundamentales van estrictamente relacionados con el análisis de los intereses de los clientes y haber establecido un público meta, ya que este objetivo puede modificar, en gran medida, las cualidades que deben cumplir estos pilares. Una vez que el negocio tiene una lista de proveedores de confianza en los cuales va a adquirir los productos, bienes, servicios o materia prima para su operación, el siguiente paso es contar

con un sistema de inventarios sólido, que le permita a la empresa tener una noción de cuáles productos se están quedando sin existencia, o bien un sitio en donde alojar información relevante del producto como descripción o precios. Es por esto por lo que Ferrero (2015) nos plantea lo siguiente:

La gestión de inventarios es un aspecto básico en todas las empresas, ya sean industriales o comerciales. En términos generales, los inventarios son el conjunto de materiales y suministros que una empresa posee, con el fin de venderlos o para abastecer el proceso productivo (Ferrero, 2015, p. 19).

Como lo indica Ferrero (2015), las operaciones de una empresa dependen completamente de la cantidad de suministro que tengan en sus almacenes para poder llevar la operación a cabo. Es fundamental dentro de una organización contar con un sistema centralizado que les permita a los encargados del manejo de inventario tener un control preciso de la cantidad de suministros que cuentan para venta, para reparaciones de equipo o bien de materia prima necesaria para la manufactura de algún producto. En relación con el prototipo propuesto para la Librería y Bazar Dieka, se busca contar con un módulo de inventario en el que se pueda llevar todo el control de existencias de los productos, así como de todos aquellos atributos que se consideren relevantes para el uso de esta información.

Cuando se habla de inventario de productos, el almacenaje de inventario no solo se limita a aquellos insumos que se encuentran a la venta o en servicio al cliente, es indispensable también contar con un inventario controlado del suministro interno que requieren los colaboradores para llevar a buen puerto, las funciones que desempeñan diariamente.

Ampliando la información con respecto al almacenaje de productos y según la investigación planteada por Ferrero (2015), los inventarios se pueden clasificar en las siguientes categorías, según la naturaleza de los suministros que manejen:

- Stocks de materias primas: Comprende todo aquello que va a formar parte del producto final.
- Stocks de materiales: Comprende aquello necesario para el proceso de fabricación que no necesariamente son apreciables en el producto final.
- Stocks de productos en proceso: Son los productos o componentes que se encuentran en proceso de fabricación.

- Stocks de productos terminados: Son los artículos o bienes finales almacenados o vendidos a un cliente de acuerdo con su pedido (p. 20).

Para efectos de la operación de la Librería y Bazar Dieka y la presente investigación, de las naturalezas mencionadas anteriormente, se trabajará únicamente con *stocks* de productos terminados, ya que las demás naturalezas están relacionadas directamente con compañías de manufactura que, para el caso de compañías de tipo “*retail*”, no aplica.

Como se ha observado en los párrafos anteriores, el control de inventarios es un proceso indispensable para el correcto funcionamiento dentro del flujo normal de ventas; sin embargo, es importante tener noción de cuáles son las ventajas directas que se obtiene al contar con este tipo de sistemas y, a su vez, que posibles desventajas podría incurrir en la organización, es por esto que Ferrero (2015) plantea algunas ventajas y desventajas del contar con un sistema controlado de almacenaje de productos para el soporte de las operaciones de una compañía:

Ventajas:

- Reducción de los tiempos de entrega.
- Incremento de la flexibilidad.
- Reducción de costes del pedido.
- Reducción de costes de ruptura de stocks.
- Reducción de costes de adquisición y producción.
- Mejora de la calidad de la producción.

Desventajas:

- Incremento del coste de almacenamiento.
- Incremento del coste financiero.
- Ocultación de problemas (la producción no se detiene y los problemas se resuelven parcialmente sin analizarlos a profundidad).
- Posible obsolescencia de lo almacenado.
- Riesgos derivados de situaciones imprevistas dentro de los inventarios (p. 21).

En otras palabras, se puede decir que las principales ventajas de contar con un correcto sistema de almacenaje yacen en el poder controlar cuando alguna existencia está cerca de llegar a su punto mínimo de existencias esto para poder actuar con rapidez y así realizar la compra respectiva a los proveedores, por otro lado, contar con un *stock* de productos puede

economizar costos, puesto que, previendo algún aumento en los costes de adquisición en estos productos, es menor el impacto al tener varias existencias en el inventario. No obstante, se podrían considerar algunas desventajas, al tener una gran cantidad de inventario almacenado, aumentando así sus costos de almacenamiento y los costos financieros que esto requiera, y adicional a una de las desventajas que podrían causar más impacto es la obsolescencia de productos. Sin embargo, es indispensable contar con un sistema de inventario centralizado para un correcto control de los productos que se pueden ofrecer al cliente, debido a que, por el desconocimiento de esto, se puede incurrir a errores graves como el incumplimiento de una promesa al cliente o fallas en el tiempo de las entregas de un pedido. Por lo tanto, nunca se puede desestimar un uso correcto del almacenaje de productos dentro de la organización.

Una vez entendiendo los aspectos básicos del negocio expuestos anteriormente en donde se plantea los cimientos de esta investigación, se entrará más a profundidad en como los sistemas de información pueden lograr o al menos ayudar a optimizar todos los procesos expuestos en los párrafos anteriores, es por esto por lo que Oz (2006) hace referencia a la utilidad de la información en la vida cotidiana:

Las personas necesitan información por muchas razones y en diversas maneras. Por ejemplo, es probable que una persona busque información de entretenimiento para observar el televisor, ir al cine, navegar por Internet, escuchar la radio y leer periódicos, revistas y libros. Sin embargo, en los negocios, las personas y las organizaciones buscan utilizar la información de manera específica para tomar decisiones sólidas y para resolver problemas: dos prácticas muy relacionadas que son la base de toda empresa exitosa (p. 41).

Con base en lo anterior, la información es un activo fundamental dentro de una organización, ya que es el conjunto recopilado de datos que, de una forma ordenada y concisa, permite a una compañía tomar decisiones con respecto a sus proyecciones de ventas, sus gastos, sus estrategias de *marketing*, o bien al estudio del mercado en el cual se desenvuelve. Todos estos valores son de vital importancia para que una empresa pueda tener competitividad dentro del mercado; sin embargo, estos valores serían sumamente complejos de conseguir si no se cuenta con un sistema de información sólido que provea de estos insumos. Dentro del marco de esta investigación, es notable la necesidad de la Librería y

Bazar Dieka de poder tener información a su disposición para la toma de decisiones, ya que, al tener los procesos de una manera manual, se torna complejo la recopilación, procesamiento e interpretación de la información.

Cuando se habla de manejo de datos e información se suelen confundir ambos conceptos y, aunque ambos tengan una estricta relación, sus definiciones no tienen el mismo significado, es por esto por lo que Oz (2006) plantea cómo poder diferenciar los datos de la información para poder entender en forma, la utilización de sistemas de información:

- Dato: Literalmente significa hecho, el cual puede ser un número, una información o una imagen.
- Información: Son los hechos o las conclusiones que tienen un significado dentro de un contexto (p. 7).

Dicho de otro modo, un dato es todo aquel atributo que caracterice un objeto o una entidad en la vida real, a manera de ejemplo, al hablar de datos que caractericen a una persona física, se pueden identificar el nombre, los apellidos, el número de cedula nacional, los datos de contacto, los datos de residencia, la nacionalidad, entre muchos otros; cada uno de estos representan rasgos que identifican a una persona; sin embargo, cuando se entrelazan cada uno de estos datos para darle identidad a una persona en específico a esta relación entre datos se le llama información. Bajo el contexto de la Librería y Bazar Dieka, hoy en día se maneja una gran cantidad de datos de una gran cantidad de objetos y entidades, como, por ejemplo, los precios, nombres y características de los productos o el total de una venta. Estos datos, al no estar esquematizados en una estructura de información, resultan complejos poder interpretarlos y exponerlos a análisis, Oz (2006) hace referencia a este proceso de transformación de datos para convertirse en información:

Los datos básicos rara vez son significativos o útiles como información. Para convertirse en información, los datos se manipulan mediante la formación de tablas, la suma, la resta, la división o cualquier otra operación que permita comprender mejor una situación (p. 7).

Se ha verificado que en la vida cotidiana tanto de las empresas como de las personas se generan datos continuamente, como, por ejemplo, el total de la factura en las compras del supermercado, la nota de un examen o el porcentaje de interés en una transacción bancaria. Todos estos mencionados anteriormente son datos que por sí solos no pueden ser

interpretados para realizar un análisis sobre determinada situación; sin embargo, cuando a estos datos se les proporciona una estructura y se relacionan entre ellos, esto puede generar un insumo que permita estudiar y tomar decisiones conforme a este recopilado de información.

Cuando existe la necesidad de convertir datos en información, es indispensable contar con un sistema que permita esta transformación, es por esto por lo que Oz (2006) expone un sistema como “una matriz de componentes que colaboran para alcanzar una meta común o varias, al aceptar entradas, procesarlas y producir salidas de una manera organizada” (p. 9). De esta forma, se observa que un sistema se diseña, con el fin de alcanzar una meta en común, o bien solventar una problemática en específico, en donde, al admitir entradas, que estas son representadas por los datos, cuando se recopilan y estructuran, es posible analizarlas en formas de salida para poder obtener información que puede ser interpretada para la toma de decisiones y solución de problemas. Con respecto a la Librería y Bazar Dieka, se espera poder tomar todos estos datos que aún no cuentan con una estructura y por medio del prototipo darles sentido a estos datos para la construcción de información que pueda ser utilizada por el comercio.

Por consiguiente, el poder obtener información por medio de un sistema que estructure los datos que se le ingresen ha permitido optimizar una gran cantidad de operaciones en la vida cotidiana, es por esto por lo que Oz (2006) menciona algunas de las transformaciones que se han vivido en el campo de la tecnología con el auge de los sistemas de información:

- La potencia de las computadoras ha aumentado enormemente al mismo tiempo que los precios de estos productos han disminuido.
- Aumentan la diversidad y sencillez de los programas por computadoras. La tendencia del mercado de software es crear cada vez sistemas más amigables con el usuario en donde este únicamente deba tener los conocimientos básicos en computación para poder utilizarlo.
- La rapidez y confiabilidad de las líneas de comunicación y el acceso a Internet se han facilitado y difundido. Esto permite que el alcance de los productos de una organización pase de tratarse de forma local, a regional o global según el tipo de

productos que ofrezca al usuario. Dando así a las organizaciones un mercado mucho más amplio de donde operar.

- Una proporción cada vez mayor de la fuerza de trabajo mundial sabe hacer uso de una computadora. Esto hace que las operaciones en una compañía sean más seguras y eficaces (p. 14).

Es por ello que no solo las operaciones del negocio perciben ventajas en la transformación de las operaciones en el uso de sistemas de información, sino que el campo de la tecnología cada día se vuelve más demandante necesitando más innovación, mejorando los procesamientos de los equipos y reduciendo costos en un mercado cada día más competitivo para poder lograr mayor captación de clientes, con esto, el más grande beneficiado resulta ser el consumidor de servicios de tecnología que recibe directamente el progreso en el campo de la tecnología. Sin embargo, a pesar de las ventajas notables que ofrece un sistema de información a las organizaciones y personas en general, todos los días surgen nuevos retos a enfrentar posibles aspectos negativos en el uso de los sistemas de información; por ello, Oz (2006) enumera algunos de los retos que en la actualidad la tecnología y los sistemas de información pueden presentar:

- Privacidad del cliente: La posibilidad de recopilar, conservar, manipular y transferir datos de manera poco costosa y rápida permite a las personas y a las organizaciones reunir millones de registros personales. Esto levanta un reto sumamente importante en las organizaciones que manipulan esta información, en donde deben asegurar a los clientes que su información será privada si él así lo requiera y esta únicamente será difundida con su consentimiento.
- Libertad de expresión: La web abre oportunidades para muchas actividades que las personas consideran indeseables, como la difusión de imágenes violentas, pornográficas y la difusión de trabajo digitalizado copiado de manera ilegal. Las organizaciones tienen que velar que sus colaboradores, utilicen los sistemas de información y los recursos que provee las tecnologías de una manera siempre desde una perspectiva ética en donde no se exponga su seguridad, ni la seguridad de la compañía, así como la información sensible de clientes que aquí se maneja.
- Molestias en línea: Este punto hace referencia a la sobreexplotación del recurso digital a los clientes en donde existen buzones de correo electrónico saturados por

publicidad o por información no deseada. Las organizaciones tienen que velar que las campañas que lancen por medio de Internet sean concisas y realizarlas de una manera calendarizada y regulada para no generar lo opuesto que se espera y crear molestia en el cliente (p. 16).

En este orden ideas, se pueden observar solo algunos de los posibles retos que puede generar la colocación masiva de sistemas de información, la proliferación de los datos en Internet y la digitalización de los procesos, en donde, gracias a estos retos, el campo de la tecnología se ha visto en la necesidad de dedicar recursos para asegurar que la información sea segura, viable y sea utilizada para el fin por el cual esta información fue sistematizada en un inicio. La seguridad de la información es uno de los puntos más críticos que incluso aun en la actualidad ciertas personas siguen teniendo resistencia al cambio y a la digitalización; sin embargo, esto no puede representar un limitante para que los sistemas de información sean la mejor alternativa, sino un reto para poder generar cada día sistemas más seguros y confiables. Por otro lado, es importante nunca dejar de lado la privacidad y las decisiones del usuario, no sobrecargarle de información, ni tampoco generar molestia por medio de correos electrónicos masivos o publicidades insistentes, la opinión y el deseo del usuario siempre es lo más importante. Para efectos del prototipo sugerido a la Librería y Bazar Dieka, estos son puntos que son tomados en alta consideración al desarrollar un prototipo que sea seguro, en donde los datos no se vean expuestos y que la información que genere el prototipo sea veraz y confiable, esto gracias a distintas estrategias como lo es el acceso restringido al sistema, la utilización de distintos roles o niveles de uso del sistema, asegurar las bases de datos, entre otras estrategias.

En adición a lo anterior, para poder entender de mejor forma cómo los sistemas de información llevan a cabo la transformación de los datos en información, es necesario comprender los distintos elementos que se ven involucrados. Los sistemas de información, a grandes rasgos, están compuestos por los siguientes componentes, según lo menciona Oz (2006):

- Datos: Una entrada que acepta el sistema para producir información.
- Hardware: Una computadora y su equipo periférico, dispositivos de entrada y salida y almacenamiento; el hardware también incluye los equipos de comunicación de datos.

- Software: Conjunto de instrucciones que le indican a la computadora como tomar los datos, como procesarlos, como presentar la información y como almacenar datos e información.
- Telecomunicaciones: El hardware y el software que facilita la transmisión y la recepción de texto, imágenes, sonidos, y animaciones en forma de datos electrónicos.
- Personas: Los profesionales y los usuarios de los sistemas de información que analizan las necesidades de información de una organización, diseñan y desarrollan sistemas de información, escriben programas de computadoras, operan el hardware, y dan mantenimiento al software.
- Procedimientos: Las reglas para lograr operaciones óptimas y seguras en el procesamiento de los datos; entre los procedimientos están las prioridades para disponer de las aplicaciones de software y las medidas de seguridad (p. 14).

En este sentido, se comprende que un sistema de información no se construye solamente por un solo elemento para poder llevar a cabo su funcionalidad, sino que lo comprenden una serie de componentes que, al entrelazarse, pueden alcanzar un objetivo en común, cuando se plantea la construcción de un sistema es de relevancia realizar una planeación detallada de cada uno de estos elementos, por ejemplo, cuáles son los datos que se van a ingresar en el sistema, qué características de *hardware* se deben tener para poder cumplir en calidad y forma el procesamiento de estos datos, a nivel de *software*, qué licenciamientos se requieren para la construcción del sistema y qué componentes a nivel de *software* se van a tener involucrados; en el ámbito de las telecomunicaciones, se tiene que determinar si el sistema según su naturaleza a cuáles otros equipos es necesario una interconexión y velar porque este enlace sea estable y con una capacidad suficiente para alojar las transacciones, en términos de personas, es necesario identificarse a cuáles personas va dirigido el sistema, ya que, de acuerdo con la población a la que vaya dirigido, van a variar aspectos de diseño y contenido del sistema de información y, por último, es necesario identificar los distintos procedimientos que se van a incluir dentro del sistema de información.

Cabe resaltar que, con la proliferación de los sistemas de información, se han generado diversidad de estructuras y de tipos de sistemas que cumplen funciones en

específico, según lo indica Oz (2006) en el siguiente párrafo, existen varios tipos de sistemas de información entre los cuales se pueden mencionar los siguientes:

- **Sistemas de procesamiento de transacciones (TPS):** La función predominante de este tipo de sistemas es de registrar los datos recopilados en los límites de las organizaciones es decir en el punto en donde las organizaciones realizan negociaciones con otras partes.
- **Sistemas de administración de cadenas de suministro (SCM):** Son los sistemas que brindan apoyo a la secuencia de actividades relacionadas con la generación de un producto o servicio.
- **Sistema de planeación de los recursos de la empresa (ERP):** Sistemas que proporcionan información que apoya la planificación de recursos, como el personal, los fondos, las materias primas y los vehículos.
- **Sistema de administración de las relaciones con los clientes:** Ayudan a administrar las relaciones de la organización con sus clientes para así mantener un control más personalizado.
- **Sistemas de Inteligencia Empresarial (BI):** Los sistemas de información cuyo propósito es obtener relaciones y tendencias de los datos básicos que pueden ayudar a las organizaciones a competir de mejor manera.
- **Sistemas de soporte de decisiones (DSS):** Sistemas que utilizan las organizaciones para por medio de fórmulas y procedimientos les facilite el proceso de toma de decisiones dentro de la organización (p. 18).

Los anteriores solo son algunos ejemplos de los distintos tipos de sistemas de información que existen actualmente en el mercado; sin embargo, estos mencionados son los más comunes de observar en las distintas organizaciones, cabe destacar que, al comprender el funcionamiento de cada tipo de sistema, se puede orientar en cómo construir una solución tecnológica para un determinado problema. En el marco del prototipo a desarrollar para la Librería y Bazar Dieka se construirá un sistema de procesamiento de transacciones, pues este sistema va a recibir todas las transacciones de compra, venta, gestión de clientes y proveedores, donde estas transacciones serán almacenadas en forma de información en una base de datos, para luego poder ser procesadas e interpretadas para distintas de operaciones,

tales como reportes de ventas, listado de productos, clientes y proveedores, entre otras operaciones.

Entender cada uno de los aspectos mencionados en este apartado en relación con los sistemas de información es de vital importancia, para poder sacar un mayor aprovechamiento de las tecnologías de información en una organización, ya que cualquier actividad manual o repetitiva puede ser transformada en un sistema de información que facilite dicha operación. Uno de los mayores aprovechamientos que se les puede dar a los sistemas de información es gracias a la tendencia en las empresas de construir y adquirir aplicaciones o servicios que sean accesibles por medio de Internet, o bien que manejen una red interna en la nube, debido a que permite el acceso desde múltiples ubicaciones ya sea fuera o dentro de la organización, de esta forma, obteniendo un sistema centralizado y, por otro lado, obtener un gran beneficio comercial, al poder vender productos a clientes de cualquier parte del mundo con solo el requisito de que la persona tenga acceso a Internet, es por esto por lo que las aplicaciones web han reemplazado en un gran porcentaje o a lo que se le conoce como aplicaciones de escritorio.

Dentro de este marco, las aplicaciones web son aquellos sistemas de información que son posibles de utilizar por medio de un navegador web y pueden ser accesibles en Internet; por lo tanto, las aplicaciones web pueden ser utilizadas desde cualquier ubicación con la única condición de que tengan un acceso a Internet. Mateu (2014) escribe la siguiente reseña que resume los orígenes de la Internet y cómo esta, de forma paulatina, se ha colocado en la vida cotidiana de las personas.

Internet, la red de redes, nace a mediados de la década de los setenta, bajo los auspicios de DARPA, la Agencia de Proyectos Avanzados para la Defensa de Estados Unidos. DARPA inició un programa de investigación de técnicas y tecnologías para unir diversas redes de conmutación de paquetes, permitiendo así a los ordenadores conectados a estas redes comunicarse entre sí de forma fácil y transparente.

A mediados de los noventa se inició el “boom” de Internet. En esa época el número de proveedores de acceso privado se disparó, permitiendo a millones de personas acceder a Internet, que a partir de ese momento ya se empezó a conocer como la Red, desbancado a las demás redes de comunicación existentes tales como Compuserve, FidoNet/BBS, etc. El punto de inflexión vino marcado por la aparición de

implementaciones de TCP/IP gratuitas (incluso de implementaciones que formaban parte del sistema operativo) así como por la popularización y abaratamiento de medios de acceso cada vez más rápidos (módems de mayor velocidad, RDSI, ADSL, cable, satélite). El efecto de todos estos cambios fue de “bola de nieve”: a medida que se conectaban más usuarios, los costes se reducían, aparecían más proveedores e Internet se hacía más atractivo y económico, con lo que se conectaban más usuarios, etc. (p. 11).

En sentido, se logra extraer que en lo que al inicio resultó como una actividad militar en busca de servir al gobierno de los Estados Unidos, el ser humano encontró la forma de poder interconectar millones de personas alrededor del mundo sin la necesidad de estar en un ámbito presencial, donde, a partir de la década de los noventa, gracias a la interconexión paulatina de una gran cantidad de redes, se empezó a ver el Internet como se conoce hoy en día, y en gran cantidad de países este recurso se ha vuelto una necesidad básica de las personas. Cada día más cosas son interconectables a Internet, ya dejó de ser un recurso únicamente dedicado a las computadoras, sino que se amplió a otros tipos de tecnologías como cámaras, consolas de videojuegos, sistemas médicos, entre otra gran cantidad de elementos tecnológicos que son completamente compatibles con el protocolo de Internet, lo cual se conoce IoT o Internet de las cosas.

Siguiendo las últimas tendencias del mercado y abriendo la posibilidad de interconectar el prototipo con otros servicios publicados en Internet, el prototipo propuesto a la Librería y Bazar Dieka será desarrollado como un servicio web, el cual se podrá colocar tanto localmente como como, incluso, publicarse en la web con todos los protocolos de seguridad necesarios para utilizarse desde cualquier ubicación, esto en caso de un crecimiento previsto en el negocio.

En relación con la idea anterior, es indispensable conocer ciertas bases teóricas, en donde se fundamentan los servicios web y que, en la actualidad, todos los lenguajes de programación se basan, Mateu (2004) menciona los siguientes elementos fundamentales dentro del campo de funcionamiento del Internet:

- La WWW o como se dice coloquialmente la web, ha dejado de ser una inmensa biblioteca de páginas estáticas para convertirse en un servicio que permite acceder a multitud de prestaciones y funciones, así como a infinidad de servicios,

programas, y tiendas. Por lo tanto, según esta definición de Carles Mateu se puede decir que la WWW es el consolidado de sitios enlazados unos con otros que permiten al usuario final consumir un servicio en específico. Es una gran red de redes que permite a todas las personas a nivel global utilizar servicios de cualquier parte del mundo.

- Protocolo HTTP: Permite una implementación simple y sencilla de sistema de comunicaciones, que permite enviar cualquier tipo de ficheros de una forma de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones, reduciendo así costos de despliegue.
- Lenguaje HTML: Se trata de un lenguaje de marcas, que permite representar de forma rica el contenido y también referenciar otros recursos, enlaces a otros documentos, o mostrar formularios para posteriormente procesarlos (pp. 12-14).

Cada uno de estos puntos mencionados anteriormente como elementos básicos tiene sus cimientos en el protocolo de Internet, siendo el WWW el conjunto de redes a nivel global interconectadas, en donde cualquier red (con los permisos adecuados) puede ser accesible desde cualquier ubicación del globo. Tal como se mencionó anteriormente, esto abre la posibilidad de que los sitios de Internet dejen de ser únicamente sitios estáticos que muestren contenido transformándolos en aplicaciones completas que pueden consumir un servicio que este alojado en cualquier parte del mundo. Por otro lado, el lenguaje HTML sigue utilizándose en la actualidad como la base de marcas de hipertexto que permite definir la estructura de los sitios web. A lo largo de los años, se han incluido más etiquetas y más funcionalidades; sin embargo, su razón de ser sigue siendo la misma para lo que fue concebido.

Por otro lado, siendo el protocolo HTTP el que hace posible el envío de información y ficheros entre distintos sitios, este contiene ciertas directivas que permiten a los sitios comunicarse entre sí, enviar peticiones y que estas sean respondidas para presentarlas al usuario de una forma amigable para que sea más fácil su interpretación. Mateu (2004) menciona las siguientes directivas, que son compuestas en el protocolo HTTP:

- GET Petición de recurso.
- POST Petición de recurso pasando parámetros.

- HEAD Petición de datos sobre recurso.
- PUT Creación o envío de recurso.
- DELETE Eliminación de recurso.
- TRACE Devuelve al origen la petición tal como se ha recibido en el receptor, para depurar errores.
- OPTIONS Sirve para comprobar las capacidades del servidor.
- CONNECT Reservado para uso en servidores intermedios capaces, de funcionar como túneles (p. 15).

En resumen, cada una de estas peticiones http observadas en el párrafo anterior permiten la comunicación entre sitios vía Internet, de los anteriores los más comunes de observar en todas las aplicaciones que se publican en la web es el método POST y el método GET, este último permite llamar otro recurso (otro sitio web u otra página dentro del mismo sitio), sin la necesidad de enviarle parámetros únicamente enviándole la petición para cumplir una serie cantidad de operaciones y obtener un resultado. Sin embargo, cuando el escenario es distinto y se necesitan enviar parámetros o valores calculados en el sitio o página actual, para que estos valores sean interpretados por el servidor destino es requerido utilizar el método POST, que permite el movimiento de parámetros entre distintos sitios. Cada uno de estos elementos es importante de conocer para poder entender el funcionamiento las aplicaciones web y tener una base sólida de cómo están constituidos.

Por consiguiente, una vez se determine el tipo de aplicación a ser programada, sea sitio web, aplicación de escritorio o móvil, uno de los aspectos básicos es determinar el paradigma de programación utilizado para la elaboración del sistema. En las primeras fases del ciclo de vida de desarrollo de un *software* se tiene que escoger la metodología de desarrollo, las tecnologías a utilizar, así como el paradigma de programación con el que se estaría desarrollando la programación. En esta investigación se estará utilizando el paradigma de desarrollo orientado a objetos, Brooch (1998) hace referencia a este en el siguiente párrafo:

El desarrollo orientado a objetos es una tecnología ya probada y se ha utilizado para construir y entregar multitud de sistemas complejos en una amplia gama de dominios de problemas. La demanda de *software* complejo crece vertiginosamente. El valor fundamental del desarrollo orientado a objetos es que libera a la mente humana para

que pueda centrar sus energías creativas sobre las partes realmente difíciles de un sistema complejo (p. 15).

Visto de otra forma, el paradigma de programación orientada a objetos busca facilitar la codificación de un sistema, en donde, por medio de la utilización de clases y conceptos de herencia y reutilización de código, vuelve posible que los programadores inviertan su tiempo en la elaboración de las operaciones más complejas de la aplicación, evitando así el tener que repetir código para realizar la misma operación una determinada cantidad de veces. Por otra parte, Cervantes et al. (2016) adicionan lo siguiente con respecto a la programación orientada a objetos:

El paradigma orientado a objetos es útil cuando el sistema se modela de forma casi análoga a la realidad, porque así se simplifica el diseño de alto nivel. Esta analogía permite que los programadores tengan más claro cuál es el papel de cada porción del programa y de los datos, lo que facilita la creación y el mantenimiento del sistema. Además, se promueve la reutilización, pues las similitudes entre objetos se programan sólo una vez en forma abstracta y el programador concentra su esfuerzo en las diferencias concretas (p. 16).

Desde esta perspectiva, la programación orientada a objetos permite un nivel de abstracción mucho mayor a un sistema de información, permitiendo asimilar el código del sistema con la realidad, de esta forma, haciendo mucho más sencilla su programación. Puede que un sistema de menor tamaño no represente un problema al repetir código distintas veces para realizar tareas distintas; sin embargo, cuando se habla de sistemas de gran tamaño que involucran gran cantidad de módulos y características, es indispensable la reutilización del código, ya que, al no hacerse de esta forma, se volvería prácticamente imposible dar un soporte adecuado al mismo. Para efectos del prototipo a desarrollar para la Librería y Bazar Dieka, este paradigma de programación se ajusta perfectamente, ya que hay una gran cantidad de operaciones que pueden repetirse en distintos módulos, por ejemplo, el consultar información de un cliente se requiere tanto para extraer reportes de los módulos de clientes, para relacionar un cliente a una venta o para observar un histórico de compras realizadas, la programación orientada a objetos permite generar una sola clase que realiza esta operación de consulta y reutilizar esta clase, en donde quiera que esta sea requerida.

Dentro de este mismo contexto aparece un elemento importante llamado objeto; que según Cervantes et al. (2016), se define como:

Una entidad virtual (o entidad de software), con datos y funciones que simulan las propiedades del objeto. Los objetos con los que se construyen los programas se ven como si fueran máquinas, las cuales están formadas por un conjunto de elementos autónomos. Las propiedades individuales de estos elementos y las relaciones entre sí definen el funcionamiento general de la máquina (p. 17).

En adición a lo anterior, en el paradigma de programación estructurado, únicamente se disponían de los tipos de datos suministrados por los lenguajes de programación; no obstante, con la creación del paradigma orientado a objetos, se dio la oportunidad de crear tipos de datos definidos por el usuario. Estos tipos de datos, también llamadas clases, permiten a los programadores crear objetos independientes con métodos especializados para dichos objetos y que puedan reutilizarse en cualquier fragmento del código. Por ejemplo, un objeto puede ser vehículo, este tiene atributos como marca, color, dimensiones, año, kilometraje entre otros atributos relacionados a este concepto. A su vez, este objeto puede contener métodos relacionados con el manejo de estos atributos, como, por ejemplo, podría ser registro de kilometraje, asignación de dueño, registro de placa, los métodos van a depender de la lógica que se requiera en la aplicación y no hay límite de la cantidad de métodos que un objeto puede contener. Dentro del prototipo a desarrollar para la presente investigación, se observan varias clases, tales como venta, cliente, proveedor, producto, usuario, entre otras, cada una de estas clases va a contener sus propios atributos definidos y sus propios métodos para poder procesar esta información.

Debe señalarse que, cuando se habla de la elaboración de un sistema, una de las capas a las cuales es de suma importancia poner atención es la de bases de datos; realizar una diagramación de entidad relación siguiendo todas las normas, hacer un estudio de *hardware* que soporte la transaccionalidad de la base de datos son algunas de las prácticas mínimas necesarias para la elaboración de una base de datos.

Las bases de datos se pueden visualizar indirectamente en cualquier aspecto de la vida cotidiana y representa uno de los componentes base para que un sistema de información pueda funcionar de manera correcta. Por ejemplo, cuando se va a realizar una transacción a un banco, los datos personales están incluidos en bases de datos de gran tamaño que

almacenan todo lo relacionado con información personal, balances de cuentas, saldos entre otros, o bien cuando se realiza una compra en un supermercado cada artículo que se compra está registrados en bases de datos que llevan el control de la existencia de estos artículos. Ricardo (2014) se refiere a las bases de datos como:

Un entorno de base de datos integrada tiene un único gran repositorio, llamado base de datos, que usa de manera simultánea muchos departamentos y usuarios en una organización. Todos los datos que la organización necesita para un grupo específico de aplicaciones, o incluso para todas sus aplicaciones, se almacenan juntos, con tan poca repetición como sea posible. Todos los accesos a la base de datos están controlados mediante un sofisticado paquete de *software* llamado sistema de gestión de base de datos (DBMS, por sus siglas en inglés). Este paquete tiene programas que establecen las estructuras de almacenamiento originales, cargan los datos, aceptan peticiones de datos de programas y usuarios, dan formato a los datos recuperados de modo que aparezcan en la forma que el programa o el usuario esperan, ocultan datos a los que un usuario particular no debe tener acceso, aceptan y realizan actualizaciones, permiten el uso concurrente de los datos sin hacer que los usuarios interfieran unos con otros, y realizan respaldos y procedimientos de recuperación automáticamente. Éstas son sólo algunas de las muchas funciones del sistema de gestión de la base de datos (p. 29).

Como lo expone Ricardo (2004), el entorno de base de datos es un componente de los sistemas de información muy complejo que, aunque se relacione de forma directa con las aplicaciones el entorno de base de datos, es un ambiente independiente que tiene su propia administración, un nivel de seguridad personalizado incluso cuenta con *softwares* especializados únicamente para el manejo y administración de las bases de datos. Dentro de una base de datos, se define una propia estructura para el almacenaje de la información en donde esta tiene que seguir una serie de normas para que la repetición de la información sea lo menor posible. Por otro lado, es necesario que los recursos que se le asignen a una base de datos sea lo más ajustado al volumen transaccional de la aplicación, ya que en esta capa es donde se perciben ocasionalmente tiempos de respuesta más prolongados, al no ser bien dimensionada una base de datos. Dentro del contexto de las bases de datos, existen varios

roles de personas según lo expuesto por Ricardo (2004), cuando se refiere a términos de bases de datos se pueden diferenciar los siguientes roles dentro de una organización:

Usuarios finales: La base de datos se diseña, crea y mantiene para satisfacer las necesidades de información de los usuarios finales, las personas que usan los datos, para realizar sus labores.

Programador de aplicaciones: Este grupo incluye a los programadores que escriben aplicaciones batch o interactivas, para otros usuarios. Los programas pueden ser escritos en una gran variedad de lenguajes de programación como lo son Java, C#, ASP.NET entre otros. De igual forma existen programadores dedicados únicamente para la programación de rutinas y procedimientos en las bases de datos.

Administradores de bases de datos: El administrador de bases de datos es la persona o grupo de personas, del diseño, creación de la estructura, y mantenimiento de la base de datos (p. 31).

Dicho de otro modo, los mencionados roles cumplen una función indispensable en el término de una base de datos, en las fases de diseño de un *software* es indispensable conocer al usuario meta para que la estructura a desarrollar en la base de datos se ajuste a todas las necesidades que tenga el negocio, cuando esta información es recolectada y analizada la persona que se encarga de desarrollar y diseñar esta estructura es el programador de aplicaciones que en algunos casos en organizaciones muy grandes es llamado programador de bases de datos que se enfatiza únicamente en el diseño y desarrollo de esta capa, en organizaciones de menor tamaño el recurso que programa el *software* normalmente es el encargado también del diseño de la base de datos, una vez que una base de datos se encuentra en producción, existe otro rol denominado administrador de bases de datos que se encarga del mantenimiento, resolución de problemas y respaldo de la información de este componente.

Otro término importante dentro del esquema de las bases de datos es el de sublenguajes de bases de datos, estos son diferentes tipos de sintaxis dentro del motor de bases de datos que se dedican a objetivos distintos. Ricardo (2004) hace referencia a los siguientes sublenguajes que existen en los entornos de bases de datos:

- DDL (Lenguaje de definición de datos): El DDL se usa para describir la base de datos.

- DML: El DML se usa para procesar en la base de datos (p. 77).

En este sentido, se comprende que el lenguaje DDL es todo aquel que se encarga de la definición de objetos dentro de la base de datos, tales como creación de tablas, procedimientos almacenados o funciones, así como el lenguaje para poder modificar o borrar estos objetos y, por otro lado, el lenguaje que permite extraer la información de los objetos de base de datos es conocido como DML, que se le puede conocer también como lenguaje de consultas. Como se observó, el lenguaje DDL permite crear objetos de base de datos, existen una gran variedad de objetos dentro de un motor de bases de datos algunos tienen el objetivo de almacenar datos, otros de ejecutar ciertas acciones dentro de la base de datos o bien existen objetos que controlan la seguridad de esta, esto son solo algunos ejemplos de la gran diversidad de elementos que se puede encontrar en una base de datos. También puede variar según la tecnología de motor de bases de datos que se utilice; sin embargo, a grandes rasgos, Ricardo (2004) menciona algunos que se puede decir que son los objetos más importantes que componen un motor de bases de datos, los cuales son:

Tablas: El modelo relacional se basa en el concepto de relación, que se representa físicamente como una tabla o arreglo bidimensional. En este modelo, las tablas se usan para contener información acerca de los objetos a representar en la base de datos. Al usar los términos del modelo entidad-relación, los conjuntos de entidades y de relaciones se muestran usando tablas.

Índices: Opcionalmente se puede crear índices para las tablas con el fin de facilitar la rápida recuperación de registros con valores específicos en una columna. Un índice sigue la pista de cuáles valores existen para la columna indexada y cuáles registros tienen dichos valores

Procedimientos almacenado o Funciones s: También denominados rutinas internas, estas facilidades se usan para escribir rutinas SQL que se puedan guardar con el esquema de la base de datos y se invoquen cuando se necesite (en contraste, los programas escritos en un lenguaje huésped se conocen como rutinas externas) (p. 149).

Estos son solo algunos de los objetos que componen el gran universo de los motores de bases de datos; sin embargo, entran dentro de la base fundamental del funcionamiento de estos motores.

Finalmente, dentro de este marco referencial se expusieron puntos clave para el mejor entendimiento del prototipo a desarrollar para la Librería y Bazar Dieka, en donde, entendiendo los principales factores del negocio, como lo son la correcta gestión de ventas, compras y almacenaje de productos, se comprende que con la utilización de un sistema de información estos procesos pueden optimizarse y reducir los posibles errores que pueden generar el fallo humano. Este sistema, al ser de naturaleza web cuenta con bastas ventajas al tener la posibilidad de interconexión con otros servicios publicados en Internet. Por otro lado, el prototipo al ser diseñado dentro del paradigma orientado a objetos se asegura que el código sea lo más optimizado posible, reduciendo la recodificación innecesaria y facilitando su interpretación por medio de un diseño de clases. Por último, se plantea un desarrollo de base de datos para poder almacenar de forma estructurada la información que genere este sistema

CAPÍTULO III: MARCO METODOLÓGICO

Enfoques de investigación

Los enfoques de una investigación corresponden al diseño que sigue el investigador para poder dar respuesta al planteamiento o problemática de la investigación. Según Hernández Sampieri (2017), “el investigador utiliza los diseños para analizar la certeza de las hipótesis formuladas en un contexto en particular o para aportar evidencias respecto de los lineamientos de la indagación (si es que no se tienen hipótesis)”(p. 97), dicho en otras palabras, los enfoques de investigación permiten realizar un planeamiento de cómo dar respuesta a las distintas hipótesis en una investigación, por ejemplo, de qué forma se obtendrá la información para llevar a cabo la investigación o cómo será procesada esta información para dar respuesta o solución al problema planteado.

Se pueden identificar tres tipos de enfoques de investigación, los cuales son cualitativo, cuantitativo y mixto. En el enfoque cuantitativo “la calidad de una investigación se relaciona con el grado en que se aplique el diseño tal cual fue concebido” (Hernández Sampieri, 2017, p. 98), este tipo de diseño se aplica principalmente en experimentos en donde se tiene que llevar a cabo un plan más riguroso en conformidad con el diseño propuesto. Por otra parte, el enfoque de investigación cualitativa permite ser más flexible, debido a que “la indagación cualitativa es en gran medida emergente y se encuentra aún más sujeta a las circunstancias de cada contexto o ambiente.” (Hernández Sampieri, 2017, p. 114), esto quiere decir que la respuesta a la hipótesis de la investigación puede cambiar según diferentes circunstancias o tipos de población, dentro del diseño se tiene que contemplar el plan o estrategia para recolectar la información que se requiera a fin de dar una respuesta al planteamiento del problema. Por último, en ciertos escenarios, se pueden observar un conjunto de ambos diseños tanto cualitativos como cuantitativos, a este tipo de enfoque se le conoce como mixto.

Enfoque cualitativo

Como se observó en el apartado anterior, el enfoque cualitativo basa una investigación en percepciones y experiencias, a continuación, se puede observar una definición de este enfoque:

En la investigación cualitativa, el diseño también implica el plan o estrategia para recolectar la información que se requiere a fin de responder al planteamiento del problema. Aunque resulta más flexible que el diseño cuantitativo en cualquier modalidad, porque la indagación cualitativa es en gran medida emergente y se encuentra aún más sujeta a las circunstancias de cada contexto o ambiente. (Hernández Sampieri, 2017, p. 114).

Como se puede observar en la definición anterior, la investigación cualitativa es ideal para orientarse en un contexto o ambiente en específico. Al ser el público objetivo en esta investigación una empresa, este enfoque de investigación se ajusta a las necesidades de esa población el particular. Hernández Sampieri (2017) agrega lo siguiente:

Asimismo, el diseño cualitativo puede subrayar un aspecto cultural, un proceso, la experiencia humana, una historia de vida o varias, la resolución de una problemática específica, el entendimiento de las relaciones entre variables o la comprensión de un caso, lo cual le otorga un sello diferente o clase de abordaje, y esto se deriva directamente del planteamiento del problema. Se escogen diseños cualitativos cuando el investigador quiere explorar, describir y conocer con amplitud y profundidad percepciones, emociones, sentimientos, experiencias, enfoques y puntos de vista de personas, desde la perspectiva de los propios participantes o sujetos investigados, en su ambiente natural y de manera más abierta (p. 114).

Dicho en otras palabras, el enfoque cualitativo permite basar la investigación desde distintas perspectivas según grupos poblacionales, esto permite una mayor amplitud para el entendimiento de una problemática que puede afectar a distintos tipos de población. Un ejemplo de este tipo investigación es el de un ingeniero industrial que requiere aumentar la productividad y eficacia de la producción y para esto realiza un estudio de las distintas tareas de los operarios de planta, esto con el fin de obtener información necesaria para plantear una solución en pro de la eficacia de las operaciones. Las unidades de análisis van a ser construidas en relación de la perspectiva de cada operario de la planta.

Por último, dentro del enfoque cualitativo, se pueden destacar las siguientes características:

1. En la investigación cualitativa no se sigue un solo camino o una ruta preestablecida, esta se va construyendo conforme se avanza el desarrollo de la investigación, dándole así la oportunidad de sufrir transformaciones según los resultados recolectados.
2. El punto de partida del investigador es la observación de la realidad, en donde por medio de la recolección de datos puede formarse una teoría de lo que ocurre en relación con la problemática.
3. Los tipos de recolección de datos pueden ser de muchas formas tanto visuales como orales, como por ejemplo entrevistas, reuniones o bien distintos tipos de documentaciones pueden permitir al investigador forjarse un criterio de la realidad.
4. Se buscan patrones o anomalías dentro en los fenómenos estudiados en donde según lo observado se pueden realizar comparaciones en dichas problemáticas presentes en distintos escenarios.
5. La mayoría de las hipótesis de la investigación se construyen según avanza la recopilación de información e interpretación de los resultados ya que es en este punto en donde el investigador se forma un criterio de la realidad y puede plantear las distintas hipótesis para la solución de la problemática.
6. La recolección de datos está basada en los distintos puntos de vista, experiencias o sentimientos del sector poblacional a ser investigado.
7. Lo que se le conoce como realidad es definida por las interpretaciones de los involucrados en la investigación es por esto por lo que, aunque ambas investigaciones tengan hipótesis similares sus resultados serán variantes por la diferencia de perspectiva de los autores.

Enfoque de investigación seleccionado

Según las características observadas en cada enfoque y posterior a un análisis minucioso sobre la problemática a solucionar, se determina que el enfoque a seguir en esta investigación es cualitativo, ya que se basa en experiencias, percepciones y, como se indicó anteriormente, se puede utilizar en la resolución de una problemática en específico. Por la naturaleza de la creación del prototipo, se busca mitigar ciertas problemáticas de Librería y Bazar Dieka. Por otro lado, la investigación no lleva un camino predefinido para su ejecución, sino que, con base en la información recolectada, se genera un criterio y con este se plantea

los distintos problemas y soluciones, llevando así un análisis exhaustivo de las problemáticas y necesidades que enfrenta el negocio, se pretende plantear un *software* que dé por solucionado estos problemas.

Método de investigación

Dentro de los métodos de investigación se encuentra la denominada investigación científica, según Hernández Sampieri (2017), la investigación científica “es una indagación o búsqueda sistemática de conocimientos que pueden ser comprobados y que tienen que ver con todo que sucede” (p. 20). Dicho de otro modo, la investigación científica busca generar interrogantes de lo que sucede en el entorno y, por medio de conocimientos y técnicas previamente estudiados (bases científicas de los hechos), se busca dar respuestas a estas interrogantes, según el criterio del investigador. En la investigación científica se observan dos tipos bien definidos, donde se encuentra la investigación científica básica e investigación científica aplicada en donde la diferenciación entre ambos tipos nace en el enfoque de la interrogante, donde la investigación científica básica plantea teorías de una forma general y abstracta y, por contraparte, la investigación científica aplicada se orienta a una problemática en específico.

Investigación científica aplicada

La investigación científica aplicada es un método de investigación en donde su naturaleza busca responder interrogantes a un problema en específico, en palabras de Hernández Sampieri (2017):

La investigación científica aplicada, en cambio, hace preguntas enfocadas en solucionar problemas específicos de un tiempo y un lugar o en generar desarrollo tecnológico. Por lo regular se basa en teorías que han sido resultado de investigación básica, solo que, como su nombre lo indica, se pone a prueba la aplicación de esa teoría en un aspecto en concreto y sus resultados son útiles para ser implementados (p. 31).

Dentro de este marco, se infiere que la investigación científica aplicada se basa de los resultados obtenidos de una investigación científica básica; por lo tanto, el conocimiento generado en este tipo de investigaciones se utiliza como base para poder resolver

problemáticas de un sector poblacional en específico sea una comunidad de personas, o una empresa. Uno de los objetivos principales de la investigación científica aplicada es generar desarrollo tecnológico en donde utilizando conocimientos científicos, se pueden transformar labores rudimentarias en procesos más optimizados utilizando la ciencia y tecnología.

La investigación científica aplicada se puede identificar por 3 características puntuales:

- La investigación científica aplicada es sistemática, esto implica que hay una disciplina detrás de la investigación realizada en donde por medio de conocimientos, teorías y métodos ya establecidos se siga un esquema ordenado para la resolución del problema. En donde los resultados de la investigación no nacen de la casualidad sino de un resultado esperado por medio de la experiencia y el conocimiento.
- La investigación científica aplicada es empírica, en otras palabras, que para llevar a cabo la investigación se recolectan y analizan datos donde por medio de esta recopilación de información se construyen las distintas hipótesis que llevan a cabo la investigación.
- La investigación científica aplicada es crítica, lo que quiere decir, que se evalúa y mejora de manera constante, no se queda estática en un solo resultado, sino que abre la posibilidad a más hipótesis para poder mejorar el resultado obtenido en la investigación.

En conclusión, la investigación científica aplicada abre la posibilidad para que las respuestas obtenidas para cada hipótesis puedan ser utilizados por la sociedad para contribución del desarrollo tecnológico y, a su vez, las propias hipótesis puedan ser mejoradas constantemente, facilitando así la transferencia de conocimientos entre distintos profesionales.

Método de investigación seleccionado

Según lo observado con respecto a los tipos de investigación científica la presente investigación busca un método de investigación aplicada para su desarrollo ya que por medio de teorías ya existentes se pretende implementar un desarrollo tecnológico en la aplicación de estas teorías con el fin de buscar soluciones a problemáticas en una empresa en específico siendo ésta la Librería y Bazar Dieka ubicada en Moravia.

Otra razón importante por la que fue seleccionado este método de investigación es debido a su propiedad de “ser empírico”, lo que permite construir la solución del problema a base de las experiencias y perspectivas del usuario que operan en la librería por medio de recursos de recolección de información tales como entrevistas o reuniones. Y, por último, otra cualidad importante al escoger este método de investigación es abrir la posibilidad a la mejora continua de la solución propuesta, permitiendo así que pueda ajustarse a las futuras necesidades que tenga el negocio.

Fuentes de información

Las fuentes de información se definen de la siguiente manera: “una fuente de información es todo aquello que nos proporciona datos para reconstruir hechos y las bases del conocimiento. Las fuentes de información son un instrumento para el conocimiento, la búsqueda y el acceso de a la información” (Maranto y González, 2015, p. 2); con base a la definición anterior, se puede decir que una fuente es todo aquel insumo de información, en lo que va a permitir el enriquecimiento y fortalecimiento de esta investigación, a continuación, se van a mencionar cuáles serán las fuentes en estudio.

Fuentes de información primaria

Las fuentes de información primarias son aquellas en las que el investigador puede consultar de primera mano para apoyar las teorías y para poder basarse en investigaciones ya existentes, con el fin de guiar la investigación, Maranto y González (2015) definen este tipo de fuente como aquellas que “contienen información original es decir son de primera mano, son el resultado de ideas, conceptos, teorías y resultados de investigaciones. Contienen información directa antes de ser interpretada, o evaluado por otra persona” (p. 3). Por lo tanto, las fuentes de información primaria suele ser el punto de partida para que el investigador pueda construir la solución. Estas fuentes pueden ser extraídas de revistas científicas, monografías, libros, publicaciones oficiales, entre otros.

Como fuente de información primaria para la presente investigación, se consultaron diversas publicaciones y libros para basar los conocimientos tecnológicos y científicos en los que se basa, algunos ejemplos son *Introducción a la programación orientada a objetos*, de los autores Cervantes, Gómez, González y García, donde se pudieron referenciar conceptos

clave de la estructura de programación a diseñar en el prototipo, o bien *Desarrollo de aplicaciones web*, del autor Mateu, de donde se extrajeron referencias para fundamentar los conceptos de desarrollo web. Estas y otras fuentes de información fueron utilizadas en esta investigación.

Fuentes de información secundaria

Las fuentes de información secundarias son aquellas en las que por medio de una fuente de información primaria fue realizado un análisis, fue interpretado y procesado por un tercero. Según Maranto y González (2015), “las fuentes de información secundaria son las que ya han procesado información de una fuente primaria. El proceso de esta información se pudo dar por una interpretación, un análisis, así como la extracción y reorganización de la información de la fuente primaria”.

Como fuentes de información secundarias se consultaron distintas publicaciones que permitieron extraer varias fuentes primarias de información, un ejemplo de fuente utilizada es *Desarrollo de un software de facturación parametrizable para empresas de venta directa por catálogo*, de los autores Navarro y Ortega.

Fuentes de información terciaria

Las fuentes de información terciarias son aquellas que se presentan como una combinación de fuentes de información primaria y secundaria, en donde se procesan y ordenan ambos tipos de fuentes, Maranto y González (2015) definen las fuentes de información terciarias como “las que recopilan fuentes de información primarias o secundarias. Estas fuentes son utilizadas para buscar datos o para obtener una idea general sobre algún tema”.

Inicialmente, se consultaron ciertos sitios web para tener una noción general de los conceptos requeridos para la construcción de esta investigación.

Unidades de análisis

Hernández Sampieri (2017) define las unidades de análisis en una investigación es un “segmento de significado. Las unidades de análisis más comunes para transcripciones son: palabras, líneas, párrafos, intervenciones de personas, páginas o cada vez que se cambia de tema” (p. 82). Como se observa en la definición planteada, una unidad de análisis es todo

aquello que puede variar su estado o su valor a criterio de los factores que componen estas unidades, las unidades de análisis tienen la característica que pueden ser medidas y analizadas para por medio de ellas obtener resultados.

Variable conceptual

La variable conceptual es aquella que contiene las definiciones extraídas de fuentes científicas previamente validadas, según Hernández Sampieri (2017), la variable conceptual “es la acordada y validada por una comunidad científica o profesional. Generalmente estas definiciones se encuentran en diccionarios especializados, páginas electrónicas con respaldo institucional y publicaciones (como artículos de revistas académicas y libros)” (p. 87). Dicho de otra forma, la variable conceptual es aquella que permite respaldar las unidades de análisis con fundamentos científicos y profesionales previamente aprobados y validados por la comunidad científica.

Variable operacional

Las variables operacionales constituyen todas aquellas tareas que requiere realizar el investigador para poder medir las distintas unidades de análisis. Hernández Sampieri (2017) define las variables operacionales de la siguiente manera: “la variable operacional especifica qué actividades u operaciones deben realizarse para medir una variable. La definición operacional dice que, para recoger datos respecto de una variable, hay que hacer esto y esto otro” (p. 95), explicado de otro modo, el conjunto de pasos definidos en una variable operacional permite poder medir y trabajar sobre las unidades de análisis.

Variable instrumental

La variable instrumental es donde se especifican los diferentes métodos o medios que se utilizarán para la medición de las unidades de análisis, Hernández Sampieri (2017) indica sobre los instrumentos “aplicar los instrumentos o sistemas de medición a las unidades, participantes o casos de la investigación representa la oportunidad de confrontar el trabajo conceptual y de planeación con la realidad”. Dicho de otro modo, los instrumentos hacen posible llevar a la realidad todo aquello estipulado en las variables conceptuales y operacionales.

Las unidades de análisis definidas para la investigación pueden observarse en la siguiente tabla:

Tabla 3. *Definición de unidades de análisis de la investigación.*

Objetivo específico	Unidades de análisis	Variable conceptual	Variable operacional	Variable instrumental
Analizar los requerimientos necesarios para la creación del prototipo funcional.	Requerimientos	“El análisis de los requerimientos da como resultado la especificación de las características operativas del software, indica la interfaz de éste y otros elementos del sistema, y establece las restricciones que limitan al software” (Pressman, 2005, p. 127)	Entrevistas. Observación.	Guía de Entrevista. Guía de Observación.
Diseñar, según los requerimientos, el planteamiento estructural del prototipo que le permita a la librería la optimización de sus procesos de facturación.	Diseño.	“El diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad.” (Pressman, 2005, p. 215)	Construcción de diagramas.	Casos de uso. Diagrama de entidad relación. Diagrama de clases. Diagrama de procesos.
Programar el prototipo en todas sus capas para el cumplimiento de los requerimientos identificados.	Prototipo.	“Primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otros iguales, o molde original con el que se fabrica.” (Salazar, 2012, p. 2).	Programación del prototipo.	Microsoft SQL Server. C# ASP.NET Core MVC

Realizar las pruebas funcionales necesarias del prototipo final.	Pruebas.	“La prueba es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistemática. Por esta razón, durante el proceso de <i>software</i> , debe definirse una plantilla para la prueba del <i>software</i> : un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos.” (Pressman, 2005, p. 384).	Pruebas funcionales.	Guía de pruebas.
---	----------	--	----------------------	------------------

Fuente: Elaboración propia.

Instrumentos de recolección de datos

La recolección de información se realizó por medio de una entrevista a los propietarios de Librería y Bazar Dieka, ubicado en La Trinidad de Moravia. La entrevista se puede definir de la siguiente manera: “consiste en una reunión presencial o virtual con la finalidad de que una persona (el entrevistador) obtenga información de parte de otra (el entrevistado)” (Hernández Sampieri, 2017, p. 164).

El tipo de entrevista a utilizarse es una entrevista semiestructurada, Hernández Sampieri (2017) define las entrevistas no estructuradas como aquellas en las “el entrevistador utiliza la guía, pero tiene mayor flexibilidad (como la libertad de cambiar el orden o introducir preguntas adicionales para precisar conceptos u obtener más información)” (p. 166). Se escoge este tipo de entrevista, ya que cabe una alta probabilidad de que en el transcurso de las reuniones con el cliente surjan nuevas interrogantes, se planteen ciertos cambios en el prototipo, o se reúnan más insumos de los que se plantearon inicialmente. En

esta entrevista busca realizar el análisis de las problemáticas actuales y con ello realizar un levantado de requerimientos que se ajusten a la solución de cada una de las problemáticas.

Otro instrumento de recolección de datos utilizado en la investigación es la observación, según Herrera (2008), “la observación permite obtener información sobre un fenómeno o acontecimiento tal y como este se produce”. Esto quiere decir que la observación no está sujeto a interpretaciones o percepciones, sino que puede evaluarse el fenómeno tal cual se produce y construir así un criterio en base de lo observado. Se escoge este método de recolección de datos para cubrir todos aquellos aspectos en los que el usuario, por desconocimiento de la materia, no sea capaz de contestar por medio de la entrevista; por lo tanto, en estos puntos se puede aplicar la observación para recolectar la información requerida.

Proceso de recolección y análisis de datos

Proceso de recolección de datos

Como se pudo observar en el apartado anterior, el instrumento de recolección de datos a utilizar para esta investigación serán las entrevistas y la observación, por lo que se busca por medio de reuniones con los propietarios de la librería entender las problemáticas actuales que enfrenta el negocio, ya sea con procesos lentos de facturación, falta de control de inventario, entre otros. Estas reuniones serán de tipo presencial, por lo cual, por medio de apuntes, se recopilará todo lo indicado por el cliente, guiando la entrevista una serie de puntos previamente definidos. Se realizarán consultas referentes a los procesos presentes, solicitando un detalle de cual es actualmente el paso a paso de estos procedimientos, se conversarán las problemáticas vigentes y, por último, se recopilarán todos los detalles puntuales que el usuario espera recibir en el prototipo así sea de forma o de funcionalidad de este.

Como segundo paso para el proceso de recolección de datos se concretará una visita al negocio con el fin de poder realizar observación en los procesos de ventas y facturación, donde se espera evaluar la forma en la que se controlan precios y existencias de productos, información de clientes y reporte de factura entregado al cliente. Por otra parte, se espera observar el proceso de compra de inventario a los proveedores y manejo de cuentas por pagar.

Como último punto, se espera analizar la ejecución actual de los cierres de caja, con el fin de conocer como el cliente lleva el conteo diario de las ventas realizadas.

Una vez teniendo el detalle de las experiencias expuestas por el usuario, se realizará un análisis de requerimientos de lo que se requiere el prototipo para así poder plantear las distintas soluciones para cada inconveniente. Dentro de este análisis de requerimientos, se plantearán los diferentes casos de uso que requieren los distintos módulos del sistema, y se espera tener un panorama general para comenzar con la construcción de diagramas. Dichos diagramas y casos de uso tienen que ser presentados en una reunión posterior con el cliente para explicarlos y ajustar cualquier detalle que quede por fuera de la revisión preliminar.

CAPÍTULO IV: ANÁLISIS DE RESULTADOS

Según lo estipulado en el proceso de recolección de datos, se procedió a realizar una visita en el negocio, en donde se realizó tanto el proceso de entrevista como el proceso de observación. Se empezó realizando la entrevista a la administradora de la Librería y Bazar Dieka, las primeras preguntas iban orientadas hacia el grado de conocimiento que tienen los usuarios que operan el negocio en términos de tecnología y sistemas de información. Según los datos recopilados se logró conocer que anteriormente el negocio nunca había utilizado un sistema de información para el manejo de facturación y ventas; sin embargo, estos sí tenían un conocimiento básico en la utilización de computadoras, ya que una de las líneas de negocio que maneja la librería son los servicios de fotocopiado, impresión y escaneos de documentos. Por lo tanto, a razón de utilización del prototipo, los usuarios ya cuentan con una noción básica en la utilización de equipo tecnológico. En ausencia de un sistema de información, el negocio ha estado trabajando con la utilización de una caja registradora para apoyar el proceso de facturación y ventas; no obstante, la caja registradora tiene muchas limitantes que pueden ser cubiertas por medio de un sistema de información.

En relación con lo anterior, se consultó en la entrevista en referencia al proceso de ventas, el administrador del negocio comentó que, al momento de una venta, esta se marca en la caja registradora, sumando manualmente por medio de códigos los artículos vendidos y como una doble validación tienen un libro donde se apuntan todas las ventas que se realizan durante el día. El ticket o factura que se le da al cliente con la venta es el que genera la caja registradora donde muestra la información del negocio, la descripción de artículos vendidos y al final del comprobante se muestra el total de la venta. A pesar de que hasta el momento la caja registradora les ha sido hasta cierto punto funcional, hay ciertas limitantes como lo son: la falta de manejo de inventarios en la registradora, falta de control de datos de clientes y proveedores, falta de un control apropiado de cuentas por pagar y cobrar y el uso de estos equipos en ocasiones puede volverse tedioso por su interfaz poco amigable con el usuario, debido a estas debilidades el proceso del control de ventas es mínimo, por lo que llevar un apunte físico en cuadernos al realizar cada venta. En relación con las ventas a crédito, el administrador afirmó que este tipo de pago no aplica para todos los clientes, es una facilidad

que se les otorga a ciertos clientes como juntas directivas de colegios, o bien clientes de confianza.

Otro aspecto importante conversado en la entrevista es el proceso de compras, en donde se logró conocer que actualmente no existe un proceso real de control y manejo de compras, ya que, cuando se realiza una compra, lo único que los administradores del negocio realizan es un archivado del documento físico de la factura esto para tener una referencia de la compra realizada; sin embargo, no se registra en ningún sistema por lo que no es posible conocer la cantidad de compras realizadas, o bien tener un control claro de las salidas de dinero y entradas de inventario. De igual forma, no existe un manejo al realizar compras a crédito, estas se llevan en control mediante un libro donde se van registrando los pagos a proveedor.

Adicional a los puntos anteriores, se consultó sobre la recopilación de información básica tanto de clientes como de proveedores, en donde el administrador del negocio confirmó que actualmente no se realiza este procedimiento; no obstante, en algunos casos se maneja un directorio telefónico de ciertos proveedores y clientes, pero, adicional a esto, no se registra ningún tipo de información relevante de estas entidades.

Otro proceso importante a considerar es el manejo de inventarios y existencias, por lo que se indagó con el administrador respecto a este procedimiento, ante lo que este comentó que el conteo de inventario se realiza una vez por año, cercano al cierre fiscal según lo estipulado por la ley; sin embargo, no cuentan con un sistema que les permite conocer la cantidad de existencia de un determinado producto en tiempo real, esto genera una problemática en el negocio, pues, cuando se acaba la existencia de un producto, estos únicamente se percatan porque ya la existencia es muy mínima o igual a cero.

En adición a lo anterior, se consultaron sobre aspectos básicos de la operación como el régimen tributario; en donde se obtuvo por respuesta que utilizan el régimen simplificado, por otro lado, la cantidad de empleados que operan en la librería es de 3; sin embargo, en época alta contratan personal adicional para suplir la creciente carga de trabajo.

Otro de los procedimientos comunes en la librería es el cierre de ventas al final de día, que es generado por medio de la caja registradora y, adicional a esto, se hace un conteo de las ventas apuntadas en el libro durante ese día, se cuenta el dinero que hay en caja y se compara este valor tanto con el cierre de la caja registradora como lo registrado en el libro de ventas.

Por último, se realiza el arqueo de caja comparando los montos de dinero contado con lo arrojado por la registradora y lo escrito en libros.

Concluyendo la entrevista, se conversó acerca de las expectativas del usuario con respecto al prototipo; en términos generales, el usuario espera un sistema sencillo de usar, y no vuelva más complejo el proceso al realizar una venta, por otro lado, que cuente al menos con las mismas funcionalidades que actualmente tiene la caja registradora y que el prototipo ayude a suplir las debilidades en el proceso tales como el registro en libros (físicos) y un mejor control del inventario al momento de hacer compras y ventas.

Una vez terminada la entrevista con el administrador del negocio, este otorgó un espacio para poder realizar el proceso de observación, en la cual se buscaba poder presenciar en tiempo real las distintas operaciones que realiza el negocio y, a su vez, poder formar un criterio propio de las distintas problemáticas que actualmente enfrentan dichas operaciones. Como punto de partida, se observó la capacidad de cómputo que tenía el negocio en términos de alojar el prototipo y se logró determinar que existe equipo con las características mínimas necesarias para poder alojar el prototipo, a su vez, cuentan con una red pequeña con acceso a Internet, el sistema operativo presente en estos equipos es Windows 11 y el negocio posee los periféricos necesarios para la utilización del sistema. Sin embargo, se observó que el negocio no cuenta con una gaveta de dinero ni impresora de tiquetes que, para utilización del prototipo en toda su funcionalidad, se recomienda.

De igual forma, se observó tanto el proceso de ventas como el proceso de compras, al momento de realizar una venta en síntesis es bastante básico en donde los usuarios tienen que conocer una cierta cantidad de códigos que ingresan en la caja registradora en donde se van sumando según el precio que tengan preconfigurados estos códigos, al final de la venta se le indica a caja registradora con cuanto paga el cliente y esta le indica el cambio que se le debe otorgar, con lo que finaliza la venta: como un paso adicional, el monto total de la venta se registra en un libro de ventas, en donde los montos aquí apuntados son sumados al final del día. En este punto, se lograron determinar algunas problemáticas en términos de la complejidad del proceso, ya que no es sencillo el ingreso de estos códigos en la caja registra; adicional a que la utilización de libro físico para el control de las ventas realizadas abre mucho la posibilidad al fallo humano.

Dentro de lo observado del proceso de compras, estas son realizadas únicamente por el administrador del negocio, los empleados no tienen la autorización para realizar estos procedimientos, cuando se realiza una compra, el administrador se encarga de calcular el precio de los productos tomando en consideración impuestos, descuentos en caso de aplicarse y la utilidad o ganancia para el negocio, luego, son etiquetados de forma física y de esta manera es como se registra el precio del producto, al finalizar este procedimiento, se archiva la factura para que sirva como referencia a futuro en caso de requerirse. En el caso de compras a crédito, se cuenta con un libro en donde se registran las cuentas por pagar y los pagos abonados a estas cuentas. En este punto se observan varias debilidades en el proceso, como lo es la falta de un sistema óptimo para el registro de las compras, debido a que, al igual que el proceso de ventas, es muy susceptible al fallo humano.

En adición al aspecto anterior, se observó el proceso de manejo de inventarios, que actualmente es inexistente, ya que el conteo de inventario se realiza únicamente una vez por año cercano del cierre fiscal; por lo tanto, esto genera una debilidad, ya que, al realizar compras a proveedores, no existe un registro de las existencias de los productos adquiridos y, de igual forma, al realizar una venta, no se disminuyen estas existencias de los productos disponibles para la venta, esto genera un desconocimiento de la cantidad real de artículos en la librería.

Al finalizar la visita de observación, se puso atención en el cierre diario de ventas, este consiste básicamente en ejecutar una función de la caja registradora que realiza la sumatoria de ventas durante el día; adicional a esto, se cuenta el dinero que haya en la caja registradora y se compara este monto con el monto mostrado en la caja registradora y lo que este apuntado en los libros donde se registra cada venta. Los formatos de entrevista se pueden encontrar en los apéndices A y B.

Concluyendo tanto la entrevista como la observación, se logró recopilar suficiente información para crear un criterio de la situación actual de la empresa y los distintos puntos débiles que existen que pueden ser optimizados por medio de la implementación del prototipo.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El presente proyecto de investigación se realizó con base en los objetivos planteados en los cuales se buscó concretar la etapa de análisis, diseño, programación, y pruebas de un prototipo para la Librería y Bazar Dieka. Para cada uno de los objetivos planteados se concluye lo siguiente:

- Se realiza el análisis de requerimientos, a partir de las necesidades observadas en el análisis de resultados, de este modo, el prototipo requiere tener los módulos de facturación, compras, inventario, cierre de caja, cuentas por cobrar, cuentas por pagar, cuentas por cobrar, análisis de datos, reportes, mantenimientos de seguridad. Cada uno de estos módulos busca dar soporte a las operaciones del negocio. Se cumple el objetivo planteado con el análisis obtenido.
- Se diseña el prototipo, con lo cual se cubre cada uno de los módulos que arrojó el análisis de requerimiento, permitiendo soportar operaciones de venta, compra de productos, manejo de créditos, control de arqueos de cajas, entre otras operaciones que son comunes en la librería. Se completa el objetivo elaborando los distintos diseños de la arquitectura del sistema.
- Se programa el prototipo utilizando el lenguaje ASP NET MVC C# y una arquitectura web; se desarrolla con una interfaz amigable, de forma tal que el usuario pueda utilizarlo sin inconvenientes. Se cumple el objetivo realizando la programación del prototipo según el análisis realizado.
- Se realiza el esquema de pruebas diseñado para velar que el prototipo otorgue los resultados esperados, solucionando cada una de las problemáticas que fueron observadas en el análisis de requerimientos. Se da por terminado el objetivo con los resultados obtenidos del esquema de pruebas.

Recomendaciones

Una vez concluido el presente trabajo de investigación, se sugiere ciertas recomendaciones con el fin de que el prototipo funcional para la gestión de ventas para Librería y Bazar Dieka pueda ser utilizado en calidad y forma, por lo tanto, que se plantea lo siguiente:

- Se recomienda la utilización del prototipo por parte de la librería para poder solventar las problemáticas que fueron identificadas en el presente proyecto de investigación. Se plantea una fecha de inicio para la implementación del prototipo para el mes de julio del año 2023, el proceso de implementación tendrá una semana de duración, la tarea estaría a cargo del administrador del negocio.
- Se sugiere una capacitación al personal de la librería para la utilización del prototipo; sin embargo, al no estar dentro del alcance del proyecto de investigación, debe contemplarse por aparte de este. En caso de realizarse, se pretende realizar dicha capacitación para el mes de julio del 2023, posterior a la puesta en producción del prototipo, se plantea una duración de un día para la capacitación y una semana de observación en caso de dudas durante el proceso. El encargado de coordinar dicha capacitación sería del administrador del negocio.
- Se aconseja adoptar los procedimientos de negocio en términos de venta y compras de productos en línea con la operación del prototipo. Esta recomendación tiene que realizarse previo a la puesta en producción del prototipo y se propone su ejecución para el mes de junio del año 2023. La duración dependería de la cantidad de procesos a ajustar, pero se estima una duración de una semana para elaborar dicho análisis, quien lo estaría realizando el administrador del negocio.
- Se insta en la adquisición de un cajón de dinero y una impresora de puntos de venta para un mejor aprovechamiento del prototipo. De igual forma que la recomendación anterior, es preferible adquirir este equipo previo a la puesta en producción y capacitación del prototipo, durante el mes de junio del año 2023, los administradores del negocio son responsables de adquirir dicho equipo.

CAPÍTULO VI: PROPUESTA

Análisis detallado del software desarrollado

A partir del análisis de los resultados obtenidos como resultado tanto de la entrevista como de la observación, se evidenció que existen algunas debilidades en el procedimiento actual de las operaciones de ventas en la librería. A razón de este análisis, se determinaron una serie de módulos que tiene que contener el prototipo para poder solventar esas debilidades de proceso. Dichos módulos son mencionados a continuación:

Módulo de facturación

Este módulo se encarga de emitir las facturas por las ventas de los clientes, dicho módulo cargará el inventario previamente definido, donde se podrán escoger los ítems vendidos. Se mostrará el valor de cada ítem y, al finalizar, la venta mostrará información totalizada de descuentos, impuestos, total bruto y neto de la venta. Este módulo también se encargará de emitir el ticket que se le proporciona al cliente con los artículos vendidos y los totales de la venta. Dentro de los procesos más relevantes de este módulo se encuentra el procesamiento de una venta, donde, al momento de emitir una factura mediante el módulo de facturación, se registra la venta en un histórico de ventas que este suministra los insumos necesarios para realizar el arqueo de caja y cierre de ventas al final de la jornada de trabajo. Por otro lado, cuando se realiza la venta de un artículo en el módulo de facturación, la cantidad de artículos vendidos tienen que ser disminuidos de la cantidad de existencias de dicho artículo en el inventario. Como procesos alternos, el módulo de facturación tiene que ser capaz de diferenciar los procedimientos de compras a crédito y compras a contado. En las compras a contado se le solicita al cliente el método de pago y, en caso de ser efectivo, la cantidad de dinero entrante y se muestra al vendedor el cambio que se le tiene que proporcionar al cliente; en el caso de ventas a crédito, estas ventas tienen que alimentar el módulo de cuentas por cobrar, sumando el monto total de la venta al saldo de la deuda actual que tenga al cliente. Como reportes del módulo, se puede encontrar el listado de ventas en donde se pueda filtrar por fechas para extraer el detalle de ventas de determinado intervalo de fechas.

Módulo de compras

Este módulo se encarga del procesamiento de compras para la adquisición de productos en el inventario. Dicho modulo tiene que ser capaz de cargar la información de los proveedores y los productos respectivos a realizar la compra. Se mostrará el valor de cada ítem, y, al finalizar, la compra mostrará información totalizada de descuentos, impuestos, total bruto y neto de la compra.

Módulo de cuentas por cobrar

Este módulo se encarga de llevar el control de las cuentas por cobrar, se registran abonos sobre los créditos de los clientes, se estipulan las fechas de pago, y envía notificaciones al cliente y al encargado del negocio sobre fechas de pago cerca a vencerse. De igual forma, el módulo está ligado directamente con el de facturación, en donde, cuando se emite una venta por concepto de venta a crédito, el monto de esta venta será sumado a la deuda actual que mantenga el cliente con la librería. En este módulo el usuario puede consultar el monto de deuda actual de un determinado cliente, la fecha de pago, el monto de la cuota, y el histórico de pagos de ese cliente. Como reportes del módulo, se puede encontrar el reporte de pagos por cliente, en donde se puede visualizar todos los pagos que ha hecho un determinado cliente a la deuda que tiene activa. Cuando se realiza un pago de una cuenta por cobrar el sistema emite un tiquete que se le proporciona al cliente como comprobante de la transacción realizada.

Módulo de control de inventarios

Este módulo se encarga de mantener un control de los artículos, es decir, la cantidad de cada producto registrado dentro de este módulo; dicha cantidad se disminuye ante ventas y se aumenta ante compras a proveedores. Cuando un artículo está llegando a su mínimo, se les envía un correo a los encargados notificando la cantidad de artículos disponibles. Se utilizarán puntos de reabastecimiento que, dicho en otras palabras, son umbrales mínimos de existencias, por lo que el negocio tiene que tomar acciones de compra para que un determinado producto no llegue a quedarse sin disponibles. Este módulo tiene que ser capaz de registrar información relevante de los productos, tales como nombre, descripción, precios y descuentos en caso de aplicarse. Como reportes que contiene el módulo, se encuentra el listado detallado de los productos que se encuentran en inventario, en donde se pueda observar la información más relevante.

Módulo de cuentas por pagar

Este módulo se encarga de calendarizar los pagos a proveedores y mantener un control detallado de las cuentas por pagar que tiene el negocio con los proveedores. Cuando un pago está próximo a efectuarse, se avisa con antelación mediante correo electrónico. En este módulo el usuario es capaz de visualizar información de crédito de los proveedores tales como monto de la deuda que tiene el negocio con el proveedor, la fecha de pago para la deuda, y poder visualizar el histórico de pagos realizados al proveedor. Como reportes que contiene el módulo, se encuentra el reporte de pagos realizados a un determinado proveedor.

Módulo de cierres de caja

Este módulo se encarga de realizar el arqueo de caja al final del día para un determinado usuario conectado al sistema, el usuario ingresa al sistema la cantidad de dinero que se tiene en caja para determinado vendedor y el sistema se encarga de sumar la cantidad de ventas en el día para poder entregar un balance entre el dinero en cajas versus el monto vendido durante el día por el vendedor. Con ello, se podrá revisar cualquier incongruencia que exista en las entradas o salidas de dinero. Una vez que la caja se cierra, ya no se puede realizar otro cierre hasta el siguiente día de operación. Como reportes del módulo, se pueden encontrar el histórico de cierres mensuales y semanales.

Módulo de análisis de datos

Este módulo permite por medio de gráficas sencillas de interpretar determinar tendencias en la venta y compra de productos que ofrece el negocio.

Módulo de mantenimientos

Este módulo se encargará de realizar el borrado, inserción, modificación, actualización de datos de proveedores, clientes, artículos, líneas de artículos, entre otros

Módulo de consultas

Este módulo se encargará de generar información proporcionada de las diferentes tablas.

Módulo de reportes

Este módulo se encargará de generar información proporcionada de las diferentes tablas y procesos, pero con un formato específico, según lo solicite el usuario. Podrá ser impreso o por pantalla.

Módulo de seguridad

Este módulo se encargará de realizar la autenticación de contraseñas y definición de perfiles.

Desarrollo del prototipo

El prototipo será desarrollado en tecnología ASP NET MVC (en un modelo de 3 capas Modelo, Vista, Controlador), con lenguaje C# utilizando Entity Framework estas tecnologías son del propietario Microsoft. El IDE utilizado para la programación, será utilizado Visual Studio 2022 versión Community Edition, este *software* es de uso gratuito y puede ser descargado desde el sitio oficial de Microsoft. A pesar de que el prototipo será utilizado en un solo sitio y no será accesible desde otras ubicaciones, se desarrolla mediante una tecnología web para dejarlo como prevista en caso de requerir alojar este en la nube para la incursión de otros sitios por medio de Internet o intranet.

Según el sitio oficial de Microsoft (2022), dentro de los requisitos mínimos de *software* necesarios para ejecutar el IDE Visual Studio 2022 se encuentran los siguientes:

Tabla 4. *Requerimientos mínimos de software Visual Studio 2022.*

Requerimiento	Especificaciones mínimas
Sistema operativo soportado	Windows 11 versión 21H2 o versiones: Home, Pro, Pro Education, Pro for Workstations, Enterprise y Education Windows 10, versión 1909 o superior: Home, Professional, Education y Enterprise. Windows Server 2022: Standard y Datacenter. Windows Server 2019: Standard y Datacenter. Windows Server 2016: Standard y Datacenter.

Fuente: Microsoft, 2022.

Análisis detallado del hardware requerido

Como fue especificado en el apartado anterior el IDE para la programación del prototipo a utilizar es Visual Studio 2022 Community Edition, según la página oficial de Microsoft (2022), los siguientes son los recursos mínimos que debe tener el computador que ejecute este IDE:

Tabla 5. *Requerimientos mínimos de hardware Visual Studio 2022*

Requerimiento	Especificaciones mínimas
Procesador	Procesador ARM64 o procesador x64 de 1,8 GHz o superior (cuatro núcleos o mejor recomendado). No se admiten procesadores ARM32.
Memoria RAM	4 GB de memoria, como mínimo. Hay muchos factores que afectan a los recursos usados, se recomiendan 16 GB de RAM para soluciones profesionales típicas.
Espacio de disco	Mínimo de 850 MB y hasta 210 GB de espacio disponible, en función de las características instaladas; las instalaciones típicas requieren entre 20 y 50 GB de espacio libre. Se recomienda instalar Windows y Visual Studio en una unidad de estado sólido (SSD) para mejorar el rendimiento.
Tarjeta de video	Tarjeta de vídeo que admita una resolución de pantalla mínima de WXGA (1366 x 768); Visual Studio funcionará mejor con una resolución de 1920 x 1080 o superior.

Fuente: Microsoft, 2022.

En el presente trabajo no entra en alcance la puesta a producción del prototipo en el negocio; sin embargo, a razón de que el cliente tome la decisión de implementarlo al ser un

negocio pequeño con pocos empleados y un volumen transaccional muy bajo tanto el código fuente como la base de datos puede colocarse en una computadora de usuario final con buenos recursos, dicho de otro modo, no es requerido instalar múltiples capas de servidores para poder alojar el sistema. Sin embargo, es importante aclarar que este análisis es basado en lo observado en el análisis de resultados; si debido al crecimiento del negocio, se colocan más cajas, más locales o múltiples ubicaciones es requerido un nuevo análisis que puede incurrir en colocar servidores en la nube, múltiples capas, balanceo de cargas, alta disponibilidad entre otros aspectos que no está contemplado dentro de los alcances del presente proyecto de investigación. Dicho esto, las características recomendadas para la computadora de usuario final para alojar y utilizar el sistema son las siguientes:

Tabla 6. *Requerimientos mínimos de hardware de prototipo funcional.*

Requerimiento	Especificaciones mínimas
Procesador	Intel CoreI5 o superior es recomendado.
Memoria	8 GB mínimos de memoria, 16 GB recomendados.
Disco duro	Disco duro SSD (Estado sólido) de mínimo 250 GB, 500 GB recomendadas.
Tarjeta de video	Tarjeta de vídeo que admita una resolución de pantalla mínima de WXGA (1366 x 768).
Dispositivos periféricos	Dispositivos periféricos básicos para la utilización de un computador, monitor, teclado y <i>mouse</i> .

Fuente: Elaboración propia.

A razón de compra de este *hardware*, el costo en el mercado para adquirir los requisitos mínimos necesarios ronda los 450 000,00 colones, este monto puede variar en relación de la casa donde se adquiera el equipo y la marca del *hardware*. Cabe resaltar que el negocio actualmente ya cuenta con un *hardware* superior al mínimo necesario para la colocación del prototipo.

Como *hardware* periférico opcional es recomendable adquirir un cajón de dinero y una impresora de punto de venta, en donde, en precios del mercado, rondan los 50 000

colones y los 120 000 colones, respectivamente. Sin embargo, se cataloga este *hardware* como opcional, pues no es indispensable su adquisición para la utilización del prototipo; sin embargo, es recomendable para un mayor aprovechamiento de este.

Análisis detallado de los elementos relacionados con las telecomunicaciones

Como se mencionó en los apartados anteriores, el presente prototipo fue diseñado a la medida para un negocio de tamaño pequeño; por lo tanto, la colocación del prototipo no será en una distribución de capas, dicho de otra forma, no se colocarán las capas web, aplicaciones y datos en servidores distribuidos. Por lo tanto, en este sentido, no es requerido la interconexión entre servidores para la comunicación del sistema.

En relación con lo anterior, se induce que el tipo de red en donde se instalará el prototipo es una red LAN, conformada únicamente por equipos de usuario final, interconectados a un “*router*” con conexión de salida a Internet. Si resulta indispensable que el equipo en donde esté alojado el sistema tenga una conexión a Internet, ya que internamente el prototipo tiene métodos que envían correos electrónicos y se requiere de una salida a Internet para que el envío de correos sea satisfactorio. El enlace de Internet recomendado para la utilización del sistema es como mínimo 30/30mbs, actualmente el negocio ya cuenta con una conexión a Internet; no obstante, a razón de adquisición de un plan de Internet, el costo mensual en el mercado ronda los 20 000 colones; sin embargo, esto puede variar según el ISP al que se le contrate el servicio.

Como se indicó en el párrafo anterior es necesario que la red LAN cuente con un router con salida a Internet, de forma normal los proveedores de servicio ya rentan este equipo al contratar un plan de Internet; por lo tanto, no es requerido la adquisición de un router adicional a este.

El protocolo utilizado en las comunicaciones del prototipo es el TCP/IP y los puertos utilizados que tienen estar abiertos tanto en los equipos de red como en el *firewall* del ordenador son, el puerto 80 y 443 correspondientes a HTTP y HTTPS sucesivamente, el puerto 1433 referente a la comunicación de SQL y, por último, el puerto 587 que hace relación al SMTP de Microsoft Exchange para el envío de correos.

Descripción detallada de base de datos o almacenamiento de información

Durante la etapa de desarrollo del prototipo, se está utilizando el motor de base de datos MSSQL Server 2022 Express Edition, esta versión de MSSQL Server es gratuita y abierta para labores de desarrollo, estudio, y aplicaciones web pequeñas que no ocupan gran cantidad de recursos para el procesamiento de la información. La siguiente tabla muestra las limitantes que tiene la versión de MSSQL Express Edition:

Tabla 7. *Características MSSQL Express Edition.*

Característica	MSSQL Express Edition
Capacidad máxima de proceso utilizada por una sola instancia: Motor de base de datos de SQL Server.	Limitada a menos de 1 <i>socket</i> o 4 núcleos
Capacidad máxima de cálculo que usa una sola instancia: <i>Analysis Services</i> o <i>Reporting Services</i>	Limitada a menos de 1 <i>socket</i> o 4 núcleos
Memoria máxima para el grupo de búferes por cada instancia de Motor de base de datos de SQL Server	1410 MB
Tamaño máximo de la base de datos relacional	10 GB

Fuente: Microsoft, 2022.

En relación con lo anterior, debido al desarrollo del prototipo, no es requerido la adquisición de una licencia de MSSQL para la elaboración de este; sin embargo, a razón de puesta en producción, se recomienda utilizar la versión MSSQL Server Standard Edition, ya que, a pesar de que la librería tiene muy poco procesamiento de transacciones, el valor de tamaño máximo de la base de datos es muy poco (10 GB); no obstante, como se ha mencionado en los apartados anteriores la puesta a producción no está incluida dentro de los alcances del presente trabajo de investigación. La siguiente tabla muestra las características de la versión MSSQL Standard Edition.

Tabla 8. *Características MSSQL Standard Edition.*

Característica	MSSQL Standard Edition
Capacidad máxima de proceso utilizada por una sola instancia: Motor de base de datos de SQL Server.	Limitada a menos de 4 <i>sockets</i> o 24 núcleos
Capacidad máxima de cálculo que usa una sola instancia: <i>Analysis Services</i> o <i>Reporting Services</i>	Limitada a menos de 4 <i>sockets</i> o 24 núcleos
Memoria máxima para el grupo de búferes por cada instancia de Motor de base de datos de SQL Server	128 GB
Tamaño máximo de la base de datos relacional	524 PB

Fuente: Microsoft, 2022.

El costo de adquirir una licencia MSSQL Standard Edition ronda los \$899usd, según el sitio oficial de Microsoft. Con respecto al *hardware* requerido para la base de datos, debido a que la librería cuenta con un procesamiento bajo, la base de datos y la aplicación se alojaron en el mismo equipo en donde se utilice el sistema.

Para la ejecución de análisis de datos se utilizará la aplicación Power BI en su versión pro cuyo costo de licencia ronda los \$10usd mensuales.

Descripción detallada del personal requerido para el uso del sistema

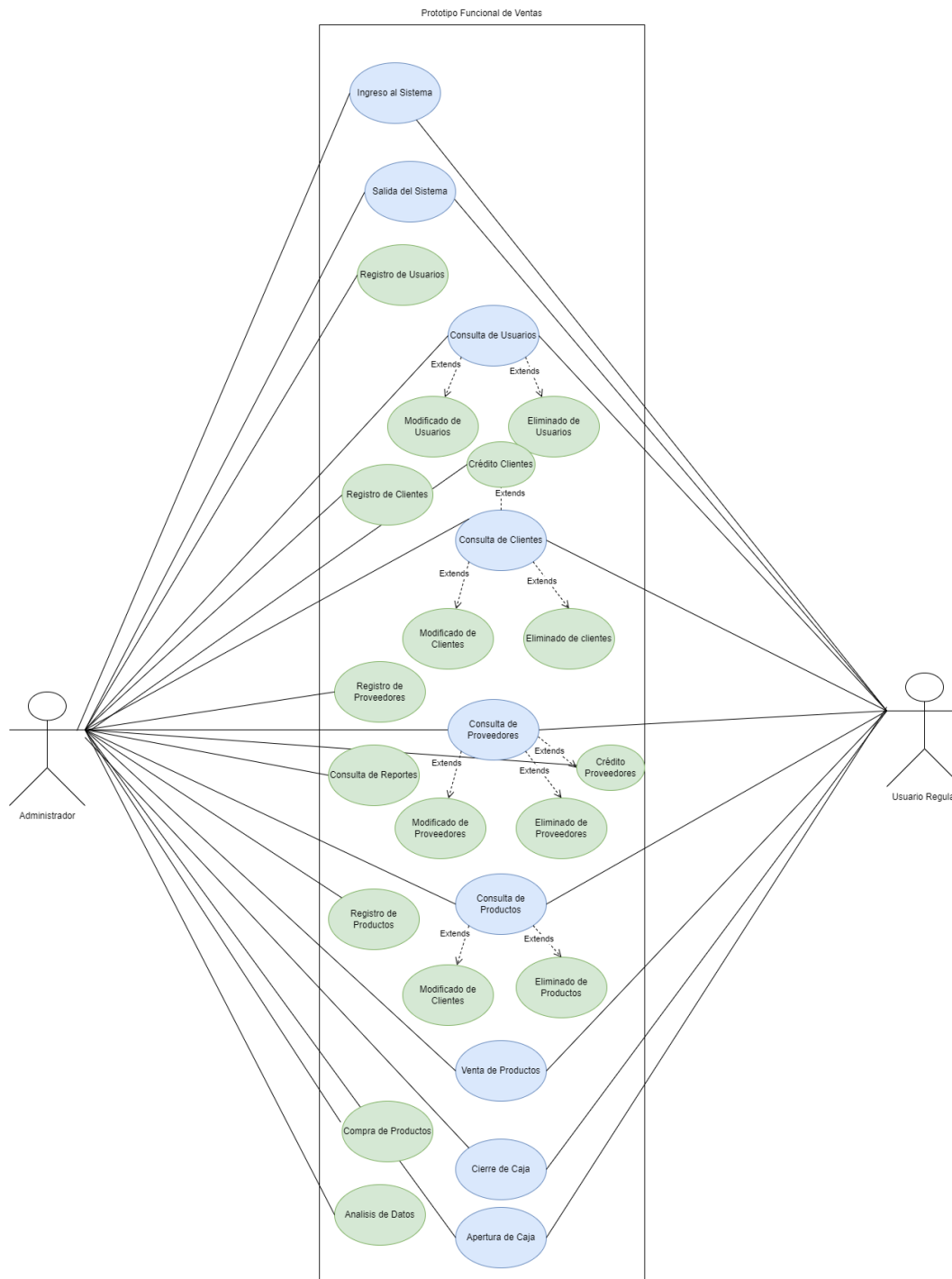
El prototipo fue diseñado de una forma tal que sea amigable y fácil de utilizar para los usuarios, esto porque a pesar de que actualmente los administradores de la librería tienen un conocimiento básico en manejo de computadoras, no cuentan con experiencias anteriores utilizando un sistema enfocado al procesamiento de ventas. Por esta razón, para la utilización del sistema no es requerido de conocimientos técnicos avanzados, o saber a profundidad temas complejos relacionados a la computación, los conocimientos requeridos son conocimientos básicos de computación, saber utilizar un navegador web, y utilizar los periféricos básicos de una computadora teclado, *mouse*, monitor e impresora.

La cantidad de personas necesarias para la utilización del sistema es como mínimo un administrador que tendrá acceso a funciones específicas como compras de productos, y mantenimiento (CRUD) de las distintas entidades del sistema (clientes, proveedores, inventario etc.) y al menos un usuario final que fungirá como operador del sistema.

La capacitación del sistema no está incluida dentro de los alcances del presente proyecto de investigación; sin embargo, es recomendable realizar dos capacitaciones, una va orientada a los administradores del sistema para mostrar los distintos procedimientos administrativos que tiene que cumplir dentro del sistema tales como agregado, modificación, eliminado; de clientes, proveedores, inventario entre otros, así mismo el mantenimiento necesario para el módulo de usuarios. Por otro lado, la segunda capacitación va orientada a los usuarios finales del sistema que operan en el negocio, en donde se muestran tareas como registro de ventas y cierres de caja, que estas son tareas competentes a sus labores dentro del negocio.

Diagrama de casos de uso

Figura 1. Diagrama de casos de uso.



Fuente: Elaboración propia.

Casos de uso

Tabla 9. *Caso de uso de ingreso al sistema.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.01
Fecha de elaboración:	30/01/2023
Descripción del caso de uso:	Ingreso al sistema
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Administrador del sistema Usuarios regulares
Precondiciones:	El usuario debe contar con un usuario previamente registrado en el sistema por el administrador.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa al URL del sitio. 2. El sistema le devuelve la pantalla de inicio de sesión donde se observan, el campo de correo electrónico, contraseña y olvido de contraseña, adicional un botón para enviar el formulario. 3. El usuario puede cambiar la contraseña, en caso de olvido de esta con base en el subflujo de cambio de contraseña. 4. El usuario digita el correo electrónico y la contraseña y, finalmente, presiona el botón para ingresar al sistema. 5. El sistema permite el acceso y muestra la pantalla principal del prototipo. 	
Subflujos	
Subflujo Cambio de contraseña	<ol style="list-style-type: none"> 1. El usuario ingresa a la opción “He olvidado mi contraseña” 2. El sistema le muestra al usuario una pantalla con un campo para colocar un correo electrónico y un botón para enviar el formulario. 3. El usuario digita una dirección de correo previamente registrada en el sistema. 4. El sistema envía un correo electrónico con un enlace para poder restablecer la contraseña. 5. El usuario ingresa al hipervínculo proporcionado por correo electrónico. 6. El sistema muestra una pantalla con un campo de contraseña, un campo de confirmación de contraseña y un botón para enviar el formulario. 7. El usuario digita la nueva contraseña, confirma la contraseña en el otro campo

	y presiona el botón para enviar el formulario. 8. El sistema redirige a la pantalla de inicio de sesión.
Flujos alternos	
Flujo alternativo credenciales inválidas	Si el usuario ingresa credenciales inválidas para ingresar al sistema, el sistema devolverá a la pantalla de ingreso de sesión y mostrará un error.
Flujo alternativo de validación	Si el usuario ingresa con un formato incorrecto el correo o la contraseña, el sistema lo devolverá a la página de inicio de sesión y le mostrará un error.
Flujo alternativo de contraseñas no coincidentes	Si el usuario al momento de cambiar la contraseña no ingresa la misma contraseña en el campo de confirmación de contraseña, el sistema lo devolverá a la pantalla de cambio de contraseña y le mostrará un error.
Requerimientos especiales	
<ol style="list-style-type: none"> 1. Los campos de contraseña en el formulario deben estar protegidos para impedir su visualización por terceros. 2. El usuario que ingrese al sistema tiene que contar con la autorización previa del administrador del negocio para la utilización del sistema. 	
Postcondiciones	
Una vez finalizado el caso de uso, el sistema establece las variables de sesión en el navegador y permite el acceso al usuario para poder utilizar las demás funcionalidades.	

Fuente: Elaboración Propia.

Tabla 10. *Caso de uso de salida del sistema.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.02
Fecha de elaboración:	30/01/2023
Descripción del caso de uso:	Salida del sistema
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Administrador del sistema Usuarios regulares
Precondiciones:	El usuario debe haber ingresado con sus credenciales previamente al sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción de Cerrar Sesión del sistema dentro del menú de navegación. 2. El sistema muestra la pantalla de inicio de sesión. 	
Subflujos	
Flujos alternos	

Requerimientos especiales
1. Es requerido que los usuarios cierren la sesión del sistema cuando este no se encuentre frente al computador.
Postcondiciones
El sistema limpia las variables de sesión en el navegador y queda disponible para un nuevo inicio de sesión.

Fuente: Elaboración propia.

Tabla 11. *Caso de uso de registro de usuarios.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.03
Fecha de elaboración:	30/01/2023
Descripción del caso de uso:	Registro de usuarios
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Administrador del sistema
Precondiciones:	El usuario debe contar con permisos de rol de administrador para ejecutar este caso de uso. El usuario debe haber ingresado previamente con sus credenciales al sistema. El usuario debe contar con la información del usuario por agregar.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario expande en la opción del menú de navegación Usuarios, seguidamente selecciona la opción del submenú Agregar usuario. 2. El sistema muestra la interfaz de Agregar usuario, esta contiene los campos de Nombre, Apellido 1, Apellido 2, Número de Cédula, Correo electrónico, Contraseña, Confirmación de contraseña y, por último, un botón para enviar los datos del formulario. 3. El usuario puede seleccionar una imagen para el usuario con base en el subflujo, cambio de imagen de usuario. 4. En caso de no ejecutar el subflujo cambio de imagen de usuario, el sistema carga una imagen por defecto al usuario. 5. El usuario ingresa los datos del usuario en el formulario, llenando los campos Nombre, Apellido 1, Apellido 2, Número de Cédula, Correo electrónico, Contraseña y Confirmación de contraseña. Por último, presiona el botón para registrar el usuario. 6. El sistema devuelve al usuario a la vista de listado de usuarios, donde podrá visualizar el usuario previamente registrado. 	
Subflujos	
Subflujo de cambio de imagen	<ol style="list-style-type: none"> 1. Dentro de la interfaz de agregar usuario, el usuario presiona el botón "Seleccionar Imagen". 2. El sistema le muestra al usuario un explorador de archivos de Windows. 3. El usuario busca la imagen a agregar en su ordenador y la selecciona con el botón abrir del explorador de archivos.

	4. El sistema le muestra al usuario el formulario con la imagen previamente cargada en el campo de imagen de usuario.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecta en los campos Nombre, Apellido 1, Apellido 2, Número de Cédula, Correo electrónico y Contraseña, el sistema lo devolverá a la pantalla de Agregar usuario y devolverá un mensaje de error en pantalla.
Flujo alternativo de contraseñas no coincidentes	En caso de que el usuario ingrese los campos contraseña y reconfirmación de contraseña con valores distintos, el sistema lo devolverá a la pantalla de agregar usuario y devolverá un mensaje de error en pantalla.
Flujo alternativo correo electrónico ya registrado	Si el usuario ingresa una dirección de correo electrónico que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de agregar usuario y mostrará un error en pantalla.
Requerimientos especiales	
1. Las contraseñas deben guardarse en formato Hash en la base de datos para que no puedan ser interceptadas por terceros no autorizados.	
Postcondiciones	
Una vez ejecutado el caso de uso, el sistema registra el usuario en la base de datos y puede ser utilizado para el ingreso al sistema mediante la interfaz de <i>login</i> .	

Fuente: Elaboración propia.

Tabla 12. Caso de uso de consulta de usuarios.

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.04
Fecha de elaboración:	30/01/2023
Descripción del caso de uso:	Consulta de usuarios
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Administrador del sistema Usuario regular
Precondiciones:	El usuario debe haber ingresado previamente con sus credenciales al sistema.
Flujo básico del caso de uso	

<ol style="list-style-type: none"> 1. El usuario expande la opción del menú de navegación Usuarios y selecciona la opción del submenú Lista de usuarios. 2. El sistema muestra un listado en forma resumida de los usuarios registrados en el sistema. 3. El usuario puede buscar un registro en específico dentro del listado con base en el subflujo búsqueda de usuario. 4. El usuario selecciona el usuario a consultar y presiona el botón Detalles de usuario. 5. El sistema muestra la información detallada del usuario. 	
Subflujos	
Subflujo búsqueda de usuario	<ol style="list-style-type: none"> 1. Dentro de la vista de listado de usuarios, en el campo de búsqueda el usuario digita ya sea el nombre, apellidos o cédula para buscar un usuario. 2. El sistema muestra en pantalla el usuario o usuarios resultantes en la búsqueda.
Flujos alternos	
Flujo alternativo de inexistencia de usuario	En el caso de que el usuario haga una búsqueda de algún registro que no coincida ningún criterio con los usuarios existentes, el sistema devuelve a la pantalla de listado de usuarios y muestra un mensaje de que no existen usuarios relacionados.
Requerimientos especiales	
<ol style="list-style-type: none"> 1. Dentro de la vista de información del usuario no se debe mostrar la contraseña de ningún usuario. 	
Postcondiciones	
Al finalizar el caso de uso, el sistema muestra el listado completo de usuarios.	

Fuente: Elaboración propia.

Tabla 13. *Caso de uso de modificación de usuarios.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.05
Fecha de elaboración:	04/02/2023
Descripción del caso de uso:	Modificación de usuarios
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	<p>El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema.</p> <p>El usuario tiene que ejecutar previamente el caso de uso CU.04.</p>
Flujo básico del caso de uso	

<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Editar dentro de la pantalla de Detalles del Usuario. 2. El sistema muestra al usuario el formulario de registro de usuario, pero con los datos precargados del usuario a modificar. 3. El usuario realiza las modificaciones respectivas sobre el registro a modificar. 4. El usuario puede modificar la imagen del registro, con base en el subflujo de cambio de imagen. 5. En caso de que el usuario no cambie la imagen, el sistema colocara la imagen existente en base de datos. 6. El usuario presiona el botón Modificar Usuario. 7. El sistema devuelve al usuario a la pantalla de Detalles para que el usuario pueda observar los cambios realizados. 	
Subflujos	
Subflujo de cambio de imagen	<ol style="list-style-type: none"> 1. Dentro de la interfaz de modificar usuario, el usuario presiona el botón “Seleccionar Imagen”. 2. El sistema le muestra al usuario un explorador de archivos de Windows. 3. El usuario busca la imagen a agregar en su ordenador y la selecciona con el botón abrir del explorador de archivos. 4. El sistema le muestra al usuario el formulario con la imagen previamente cargada en el campo de imagen de usuario.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecto en los campos nombre, apellido1, apellido 2, número de cédula y correo electrónico, el sistema lo devolverá a la pantalla de Modificar usuario y devolverá un mensaje de error en pantalla.
Flujo alternativo correo electrónico ya registrado	Si el usuario ingresa una dirección de correo electrónico que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de agregar usuario y mostrará un error en pantalla.
Requerimientos especiales	
Ningún usuario, inclusive el usuario administrador, puede realizar el cambio de contraseña de ningún usuario por regulación de seguridad, en caso de ser requerido el cambio de contraseña de un usuario se debe ejecutar el caso de uso de olvido de contraseña.	
Postcondiciones	
Una vez ejecutado el caso de uso, el sistema guarda los registros modificados en base de datos y devuelve el usuario a la pantalla de detalles de usuario para revisar la información modificada.	

Fuente: Elaboración propia.

Tabla 14. *Caso de uso de eliminado de usuarios.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.06
Fecha de elaboración:	04/02/2023
Descripción del caso de uso:	Eliminado de usuario
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema. El usuario tiene que ejecutar previamente el caso de uso CU.04.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Eliminar dentro de la pantalla de Detalles del Usuario. 2. El sistema muestra una ventana emergente solicitando confirmación si realmente se desea eliminar el registro. 3. El usuario confirma que desea eliminar el registro. 4. El sistema muestra un resumen del usuario a borrar y vuelve a pedir una confirmación si desea eliminar el registro mostrado en pantalla. 5. El usuario confirma que desea eliminar el registro. 6. El sistema devuelve al usuario a la pantalla de listado de usuarios donde ya no se mostrará el registro borrado. 	
Subflujos	
Flujos alternos	
Flujo alternativo de confirmación de Borrado	Si el usuario ingreso a la opción de borrado por error y le indica al sistema que no quiere borrar el usuario, el sistema no realiza ninguna acción y devuelve a la pantalla de Detalles de usuario.
Requerimientos especiales	
Para cuidar la integridad referencial en la base de datos, el borrado del registro se debe realizar a nivel lógico y no físico.	
Postcondiciones	
Al finalizar la ejecución del caso de uso, el sistema realiza la modificación de la columna de habilitado por el valor de "False" para así procesar el borrado lógico del registro.	

Fuente: Elaboración propia.

Tabla 15. *Caso de uso de registro de clientes.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.07
Fecha de elaboración:	04/02/2023

Descripción del caso de uso:	Registro de clientes
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	<p>El usuario debe contar con permisos de rol de administrador para ejecutar este caso de uso.</p> <p>El usuario debe haber ingresado previamente con sus credenciales al sistema.</p> <p>El usuario debe contar con la información del cliente por agregar.</p>
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario expande en la opción del menú de navegación Clientes, seguidamente selecciona la opción del submenú Agregar cliente. 2. El sistema muestra la interfaz de Agregar cliente en donde esta contiene los campos de Nombre, Apellido 1, Apellido 2, Tipo de identificación, Cédula, Correo electrónico, Número de teléfono, Provincia, Cantón, Distrito, Dirección exacta, Disponibilidad de crédito y, por último, un botón para enviar los datos del formulario. 3. El usuario puede seleccionar una imagen para el cliente con base en el subflujo, cambio de imagen de cliente. 4. En caso de no ejecutar el subflujo cambio de imagen de cliente, el sistema carga una imagen por defecto al cliente. 5. El usuario ingresa los datos del cliente en el formulario, llenando los campos Nombre, Apellido 1, Apellido 2, Tipo de identificación, Cédula, Correo electrónico, Número de teléfono, Provincia, Cantón, Distrito, Dirección exacta y Disponibilidad de crédito. Por último, presiona el botón para registrar el cliente. 6. El sistema devuelve al usuario a la vista de listado de clientes, donde podrá visualizar el cliente previamente registrado. 	
Subflujos	
Subflujo de cambio de imagen	<ol style="list-style-type: none"> 1. Dentro de la interfaz de agregar cliente, el usuario presiona el botón “Seleccionar Imagen”. 2. El sistema le muestra al usuario un explorador de archivos de Windows. 3. El usuario busca la imagen a agregar en su ordenador y la selecciona con el botón abrir del explorador de archivos. 4. El sistema le muestra al usuario el formulario con la imagen previamente cargada en el campo de imagen de cliente.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecto en los campos nombre, apellido 1, apellido 2, tipo de identificación, cédula, correo electrónico, número de teléfono, provincia, cantón, distrito, dirección exacta, disponibilidad de crédito, el sistema lo devolverá a la pantalla de Agregar cliente y devolverá un mensaje de error en pantalla.
Flujo alternativo correo electrónico ya registrado	Si el usuario ingresa una dirección de correo electrónico que ya fue previamente ingresada en

	el sistema, devolverá al usuario a la pantalla de agregar usuario y mostrará un error en pantalla.
Flujo alternativo de número de cédula ya registrada	Si el usuario ingresa un número de cédula que ya fue previamente ingresado en el sistema, devolverá al usuario a la pantalla de agregar cliente y mostrará un error en pantalla.
Requerimientos especiales	
Los campos de provincia, cantón y distrito deben estar enlazados entre sí para únicamente mostrar los cantones para una provincia seleccionada, y los distritos para un cantón seleccionado. El sistema debe permitir los formatos de cédula nacional, jurídica, DIMEX e Identificación tributaria especial.	
Postcondiciones	
Una vez ejecutado el caso de uso, el sistema registra el cliente en la base de datos y puede ser utilizado en los módulos de facturación.	

Fuente: Elaboración propia.

Tabla 16. *Caso de uso de consulta de clientes.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.08
Fecha de elaboración:	04/02/2023
Descripción del caso de uso:	Consulta de clientes
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador Usuario Regular
Precondiciones:	El usuario debe haber ingresado previamente con sus credenciales al sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario expande la opción del menú de navegación Clientes y selecciona la opción del submenú Lista de Clientes. 2. El sistema muestra un listado en forma resumida de los clientes registrados en el sistema. 3. El usuario puede buscar un registro en específico dentro del listado con base en el subflujo búsqueda de cliente. 4. El usuario selecciona el cliente a consultar y presiona el botón Detalles de cliente. 5. El sistema muestra la información detallada del cliente. 	
Subflujos	
Subflujo búsqueda de cliente	<ol style="list-style-type: none"> 1. Dentro de la vista de listado de clientes, en el campo de búsqueda, el usuario digita ya sea el nombre, apellidos o cédula para buscar un cliente. 2. El sistema muestra en pantalla el cliente o clientes resultantes en la búsqueda.
Flujos alternos	

Flujo alternativo de inexistencia de cliente	En el caso de que el usuario haga una búsqueda de algún registro que no coincida ningún criterio con los clientes existentes. El sistema devuelve a la pantalla de listado de clientes y muestra un mensaje de que no existen clientes relacionados.
Requerimientos especiales	
No aplica.	
Postcondiciones	
Al finalizar el caso de uso, el sistema muestra el listado completo de clientes.	

Fuente: Elaboración propia.

Tabla 17. *Caso de uso de modificación de cliente.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.09
Fecha de elaboración:	04/02/2023
Descripción del caso de uso:	Modificar cliente
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema. El usuario tiene que ejecutar previamente el caso de uso CU.08.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Editar dentro de la pantalla de Detalles del Cliente. 2. El sistema muestra al usuario el formulario de registro de cliente, pero con los datos precargados del cliente a modificar. 3. El usuario realiza las modificaciones respectivas sobre el registro a modificar. 4. El usuario puede modificar la imagen del registro, con base en el subflujo de cambio de imagen. 5. En caso de que el usuario no cambie la imagen, el sistema colocara la imagen existente en base de datos. 6. El usuario presiona el botón Modificar Cliente. 7. El sistema devuelve al usuario a la pantalla de Detalles para que el usuario pueda observar los cambios realizados. 	
Subflujos	
Subflujo de cambio de imagen	<ol style="list-style-type: none"> 1. Dentro de la interfaz de modificar cliente, el usuario presiona el botón “Seleccionar Imagen”. 2. El sistema le muestra al usuario un explorador de archivos de Windows. 3. El usuario busca la imagen a agregar en su ordenador y la selecciona con el botón abrir del explorador de archivos.

	4. El sistema le muestra al usuario el formulario con la imagen previamente cargada en el campo de imagen de cliente.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecto en los campos nombre, apellido 1, apellido 2, tipo de Identificación, cédula, correo electrónico, número de teléfono, provincia, cantón, distrito, dirección exacta, disponibilidad de crédito, el sistema lo devolverá a la pantalla de Modificar cliente y devolverá un mensaje de error en pantalla.
Flujo alternativo correo electrónico ya registrado	Si el usuario ingresa una dirección de correo electrónico que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de agregar usuario y mostrará un error en pantalla.
Flujo alternativo de número de cédula ya registrada	Si el usuario ingresa un número de cédula que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de modificar cliente y mostrará un error en pantalla.
Requerimientos especiales	
Los campos de provincia, cantón y distrito deben estar enlazados entre sí para únicamente mostrar los cantones para una provincia seleccionada, y los distritos para un cantón seleccionado. El sistema debe permitir los formatos de cédula nacional, jurídica, DIMEX e Identificación tributaria especial.	
Postcondiciones	
Una vez ejecutado el caso de uso, el sistema guarda los registros modificados en base de datos y devuelve el usuario a la pantalla de detalles de cliente para revisar la información modificada.	

Fuente: Elaboración propia.

Tabla 18. *Caso de uso de eliminación de cliente.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.10
Fecha de elaboración:	02/04/2023
Descripción del caso de uso:	Eliminación de cliente
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema.

	El usuario tiene que ejecutar previamente el caso de uso CU.08.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Eliminar dentro de la pantalla de Detalles del Cliente. 2. El sistema muestra una ventana emergente solicitando confirmación si realmente se desea eliminar el registro. 3. El usuario confirma que desea eliminar el registro. 4. El sistema muestra un resumen del cliente a borrar y vuelve a pedir una confirmación si desea eliminar el registro mostrado en pantalla. 5. El usuario confirma que desea eliminar el registro. 6. El sistema devuelve al usuario a la pantalla de listado de clientes donde ya no se mostrará el registro borrado. 	
Subflujos	
Flujos alternos	
Flujo alternativo de confirmación de Borrado	Si el usuario ingreso a la opción de borrado por error y le indica al sistema que no quiere borrar el usuario, el sistema no realiza ninguna acción y devuelve a la pantalla de Detalles de usuario.
Requerimientos especiales	
Para cuidar la integridad referencial en la base de datos, el borrado del registro se debe realizar a nivel lógico y no físico.	
Postcondiciones	
Al finalizar la ejecución del caso de uso, el sistema realiza la modificación de la columna de habilitado por el valor de "False" para así procesar el borrado lógico del registro.	

Fuente: Elaboración propia.

Tabla 19. *Caso de uso de consulta de crédito de cliente.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.11
Fecha de elaboración:	04/02/2023
Descripción del caso de uso:	Consulta de crédito de cliente
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Final Usuario Administrador
Precondiciones:	El usuario debe haber ingresado previamente con sus credenciales al sistema. El usuario debe ejecutar previamente el caso de uso CU.08
Flujo básico del caso de uso	

<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Crédito de Cliente, dentro de la interfaz de detalles de clientes. 2. El sistema muestra al usuario la información crediticia del cliente con la información de total de deuda, monto de cuota, fecha de pago, y una tabla con el histórico de pagos realizados. 3. El usuario puede procesar un pago a la deuda, con base en el subflujo Pago de Crédito. 4. El usuario puede filtrar los pagos realizados, con base en el subflujo Filtro de pagos de Cliente. 	
Subflujos	
Subflujo pago de crédito	<ol style="list-style-type: none"> 1. En la interfaz de consulta de crédito el usuario digita el monto a pagar. 2. El sistema muestra el cambio a devolver al cliente en caso de que el monto pagado sea mayor a la cuota. 3. El usuario presiona el botón de procesar pago. 4. El sistema carga nuevamente la página de crédito de cliente y muestra en la tabla de pagos de cliente, con el pago recién realizado.
Subflujo de búsqueda de pagos	<ol style="list-style-type: none"> 1. El usuario digita una fecha en los campos fecha inicial y fecha final y luego presiona el botón filtrar. 2. El sistema carga nuevamente la página de crédito de cliente, con los datos de pagos filtrados por fecha en la tabla de pagos.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecto en el campo de pago, el sistema lo devolverá a la pantalla de Modificar cliente y devolverá un mensaje de error en pantalla.
Flujo alternativo de validación de cuota	Si el usuario procede a pagar la cuota de un crédito y el monto digitado es menor al monto de la cuota, el sistema lo devolverá a la pantalla de crédito y le mostrará un mensaje de error que el pago es menor a la cuota.
Flujo alternativo de deuda en cero	Si el usuario realiza un pago sobre un crédito que su total de deuda es igual a cero, el sistema no realiza ninguna acción y devuelve al usuario a la pantalla de crédito de proveedor, devolviendo un mensaje de error.
Flujo alternativo de validación de fechas	Si el usuario coloca un valor mayor en la fecha inicial que la fecha final, el sistema devolverá al usuario a la página de crédito de clientes y mostrará un error.
Requerimientos especiales	
La Librería y Bazar Dieka actualmente no maneja un procedimiento de intereses sobre crédito de clientes por lo tanto únicamente es requerido el control de pagos y cuotas.	
Postcondiciones	

Una vez procesado un pago en el crédito de un cliente, este se guarda en una tabla histórica en base de datos para que pueda ser consultado posteriormente. Adicionalmente, se hacen las disminuciones respectivas al monto de la deuda y la modificación de la siguiente fecha de pago del cliente.

Fuente: Elaboración propia.

Tabla 20. *Caso de uso de registro de proveedores.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.12
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Registro de proveedores
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	El usuario debe contar con permisos de rol de administrador para ejecutar este caso de uso. El usuario debe haber ingresado previamente con sus credenciales al sistema. El usuario debe contar con la información del proveedor por agregar.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario expande en la opción del menú de navegación Proveedores, seguidamente selecciona la opción del submenú Agregar proveedor. 2. El sistema muestra la interfaz de Agregar proveedor en donde esta contiene los campos de Nombre de proveedor, Cédula jurídica, Correo electrónico, Número de teléfono, Provincia, Cantón, Distrito, Dirección exacta y, por último, un botón para enviar los datos del formulario. 3. El usuario puede seleccionar una imagen para el proveedor con base en el subflujo, cambio de imagen de proveedor. 4. En caso de no ejecutar el subflujo cambio de imagen de proveedor, el sistema carga una imagen por defecto al proveedor. 5. El usuario ingresa los datos del proveedor en el formulario, llenando los campos Nombre de Proveedor, Cédula jurídica, correo electrónico, número de teléfono, provincia, cantón, distrito y dirección exacta. Por último, presiona el botón para registrar el proveedor. 6. El sistema devuelve al usuario a la vista de listado de proveedores, donde podrá visualizar el proveedor previamente registrado. 	
Subflujos	
Subflujo de cambio de imagen	<ol style="list-style-type: none"> 1. Dentro de la interfaz de agregar proveedor, el usuario presiona el botón "Seleccionar Imagen". 2. El sistema le muestra al usuario un explorador de archivos de Windows. 3. El usuario busca la imagen a agregar en su ordenador y la selecciona con el botón abrir del explorador de archivos. 4. El sistema le muestra al usuario el formulario con la imagen previamente cargada en el campo de imagen de Proveedor.

Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecta en los campos de Nombre de proveedor, Cédula jurídica, Correo electrónico, Número de teléfono, Provincia, Cantón, Distrito y Dirección exacta, el sistema lo devolverá a la pantalla de Agregar proveedor y devolverá un mensaje de error en pantalla.
Flujo alternativo correo electrónico ya registrado	Si el usuario ingresa una dirección de correo electrónico que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de agregar proveedor y mostrará un error en pantalla.
Flujo alternativo de número de cédula jurídica ya registrada	Si el usuario ingresa un número de cédula jurídica que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de agregar proveedor y mostrará un error en pantalla.
Requerimientos especiales	
Los campos de provincia, cantón y distrito deben estar enlazados entre sí para únicamente mostrar los cantones para una provincia seleccionada, y los distritos para un cantón seleccionado.	
Postcondiciones	
Una vez ejecutado el caso de uso, el sistema registra el proveedor en la base de datos y puede ser utilizado en los módulos de compras.	

Fuente: Elaboración propia.

Tabla 21. *Caso de uso de consulta de proveedores.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.13
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Consulta de proveedores
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador Usuario Regular
Precondiciones:	El usuario debe haber ingresado previamente con sus credenciales al sistema.
Flujo básico del caso de uso	

<ol style="list-style-type: none"> 1. El usuario expande la opción del menú de navegación Proveedores y selecciona la opción del submenú Lista de Proveedores. 2. El sistema muestra un listado en forma resumida de los proveedores registrados en el sistema. 3. El usuario puede buscar un registro en específico dentro del listado con base en el subflujo búsqueda de proveedor. 4. El usuario selecciona el proveedor a consultar y presiona el botón Detalles de proveedor. 5. El sistema muestra la información detallada del proveedor. 	
Subflujos	
Subflujo búsqueda de proveedor	<ol style="list-style-type: none"> 1. Dentro de la vista de listado de proveedores, en el campo de búsqueda, el usuario digita ya sea el nombre de proveedor o cédula jurídica para buscar un proveedor. 2. El sistema muestra en pantalla el proveedor o proveedores resultantes en la búsqueda.
Flujos alternos	
Flujo alternativo de inexistencia de proveedor	En el caso de que el usuario haga una búsqueda de algún registro que no coincida ningún criterio con los proveedores existentes, el sistema devuelve a la pantalla de listado de proveedores y muestra un mensaje de que no existen proveedores relacionados.
Requerimientos especiales	
No aplica.	
Postcondiciones	
Al finalizar el caso de uso, el sistema muestra el listado completo de proveedores.	

Fuente: Elaboración propia.

Tabla 22. *Caso de uso de modificación de proveedores.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.14
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Modificación de Proveedores
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	<p>El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema.</p> <p>El usuario tiene que ejecutar previamente el caso de uso CU.13.</p>
Flujo básico del caso de uso	

<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Editar dentro de la pantalla de Detalles del Proveedor. 2. El sistema muestra al usuario el formulario de registro de proveedor, pero con los datos precargados del proveedor a modificar. 3. El usuario realiza las modificaciones respectivas sobre el registro a modificar. 4. El usuario puede modificar la imagen del registro, con base en el subflujo de cambio de imagen. 5. En caso de que el usuario no cambie la imagen, el sistema colocara la imagen existente en base de datos. 6. El usuario presiona el botón Modificar proveedor. 7. El sistema devuelve al usuario a la pantalla de Detalles para que el usuario pueda observar los cambios realizados. 	
Subflujos	
Subflujo de cambio de imagen	<ol style="list-style-type: none"> 1. Dentro de la interfaz de modificar proveedor, el usuario presiona el botón “Seleccionar Imagen”. 2. El sistema le muestra al usuario un explorador de archivos de Windows. 3. El usuario busca la imagen a agregar en su ordenador y la selecciona con el botón abrir del explorador de archivos. 4. El sistema le muestra al usuario el formulario con la imagen previamente cargada en el campo de imagen de proveedor.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecta en los Nombre de proveedor, Cédula jurídica, Correo electrónico, Número de teléfono, Provincia, Cantón, Distrito y Dirección exacta, el sistema lo devolverá a la pantalla de Agregar proveedor y devolverá un mensaje de error en pantalla.
Flujo alternativo correo electrónico ya registrado	Si el usuario ingresa una dirección de correo electrónico que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de agregar proveedor y mostrará un error en pantalla.
Flujo alternativo de número de cedula jurídica ya registrada	Si el usuario ingresa un número de cédula jurídica que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de agregar proveedor y mostrará un error en pantalla.
Requerimientos especiales	
Los campos de provincia, cantón y distrito deben estar enlazados entre sí para únicamente mostrar los cantones para una provincia seleccionada, y los distritos para un cantón seleccionado.	
Postcondiciones	

Una vez ejecutado el caso de uso, el sistema guarda los registros modificados en base de datos, y devuelve el usuario a la pantalla de detalles de proveedor para revisar la información modificada.

Fuente: Elaboración propia.

Tabla 23. *Caso de uso de eliminación de proveedor.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.15
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Eliminación de proveedor
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema. El usuario tiene que ejecutar previamente el caso de uso CU.13.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Eliminar dentro de la pantalla de Detalles del Proveedor. 2. El sistema muestra una ventana emergente solicitando confirmación si realmente se desea eliminar el registro. 3. El usuario confirma que desea eliminar el registro. 4. El sistema muestra un resumen del proveedor a borrar y vuelve a pedir una confirmación si desea eliminar el registro mostrado en pantalla. 5. El usuario confirma que desea eliminar el registro. 6. El sistema devuelve al usuario a la pantalla de listado de proveedores donde ya no se mostrará el registro borrado. 	
Subflujos	
Flujos alternos	
Flujo alternativo de confirmación de Borrado	Si el usuario ingreso a la opción de borrado por error y le indica al sistema que no quiere borrar el proveedor, el sistema no realiza ninguna acción y devuelve a la pantalla de Detalles de proveedor.
Requerimientos especiales	
Para cuidar la integridad referencial en la base de datos, el borrado del registro se debe realizar a nivel lógico y no físico.	
Postcondiciones	
Al finalizar la ejecución del caso de uso, el sistema realiza la modificación de la columna de habilitado por el valor de "False" para así procesar el borrado lógico del registro.	

Fuente: Elaboración propia.

Tabla 24. *Caso de uso de consulta de crédito de proveedor.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.16
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Consulta de crédito de proveedor
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema. El usuario tiene que ejecutar previamente el caso de uso CU.13.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Crédito de proveedor, dentro de la interfaz de detalles de proveedores. 2. El sistema muestra al usuario la información crediticia del proveedor con la información de total de deuda, monto de cuota, fecha de pago, y una tabla con el histórico de pagos realizados. 3. El usuario puede procesar un pago a la deuda, con base en el subflujo Pago de Crédito. 4. El usuario puede filtrar los pagos realizados, con base en el subflujo Filtro de pagos de Proveedor. 	
Subflujos	
Subflujo pago de crédito	<ol style="list-style-type: none"> 1. En la interfaz de consulta de crédito, el usuario digita el monto a pagar. 2. El sistema muestra el cambio a devolver al cliente en caso de que el monto pagado sea mayor a la cuota. 3. El usuario presiona el botón de procesar pago. 4. El sistema carga nuevamente la página de crédito de proveedor y muestra en la tabla de pagos de cliente, con el pago recién realizado.
Subflujo de búsqueda de pagos	<ol style="list-style-type: none"> 1. El usuario digita una fecha en los campos fecha inicial y fecha final y luego presiona el botón filtrar. 2. El sistema carga nuevamente la página de crédito de proveedor, con los datos de pagos filtrados por fecha en la tabla de pagos.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecto en el campo de pago, el sistema lo devolverá a la pantalla de Crédito de proveedor y devolverá un mensaje de error en pantalla.
Flujo alternativo de validación de cuota	Si el usuario procede a pagar la cuota de un crédito y el monto digitado es menor al monto de la cuota, el sistema lo devolverá a la pantalla de crédito y le mostrará un mensaje de error que el pago es menor a la cuota.

Flujo alternativo de deuda en cero	Si el usuario realiza un pago sobre un crédito que su total de deuda es igual a cero, el sistema no realiza ninguna acción y devuelve al usuario a la pantalla de crédito de proveedor, devolviendo un mensaje de error.
Flujo alternativo de validación de fechas	Si el usuario coloca un valor mayor en la fecha inicial que la fecha final, el sistema devolverá al usuario a la página de crédito de proveedores y mostrará un error.
Requerimientos especiales	
No aplica.	
Postcondiciones	
Una vez procesado un pago en el crédito de un proveedor, este se guarda en una tabla histórica en base de datos para que pueda ser consultado posteriormente. Adicionalmente, se hacen las disminuciones respectivas al monto de la deuda y la modificación de la siguiente fecha de pago al proveedor.	

Fuente: Elaboración propia.

Tabla 25. *Caso de uso de registro de productos.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.17
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Registro de productos
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador.
Precondiciones:	El usuario debe contar con permisos de rol de administrador para ejecutar este caso de uso. El usuario debe haber ingresado previamente con sus credenciales al sistema. El usuario debe contar con la información del producto por agregar.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario expande en la opción del menú de navegación Inventario, seguidamente selecciona la opción del submenú Agregar producto. 2. El sistema muestra la interfaz de Agregar producto en donde esta contiene los campos de Nombre de Producto, Descripción, Código, Cantidad, Precio de costo, Precio de venta, Porcentaje de descuento, Tipo de producto y, por último, un botón para enviar los datos del formulario. 3. El usuario puede seleccionar una imagen para el producto con base en el subflujo, cambio de imagen de producto. 4. En caso de no ejecutar el subflujo cambio de imagen de producto, el sistema carga una imagen por defecto al producto. 5. El usuario ingresa los datos del cliente en el formulario, llenando los Nombre de producto, Descripción, Código, Cantidad, Precio de costo, Precio de venta, Porcentaje de descuento y Tipo de producto. Por último, presiona el botón para registrar el producto. 	

6. El sistema devuelve al usuario a la vista de listado de productos, donde podrá visualizar el producto previamente registrado.	
Subflujos	
Subflujo de cambio de imagen	<ol style="list-style-type: none"> 1. Dentro de la interfaz de agregar producto, el usuario presiona el botón “Seleccionar Imagen”. 2. El sistema le muestra al usuario un explorador de archivos de Windows. 3. El usuario busca la imagen a agregar en su ordenador y la selecciona con el botón abrir del explorador de archivos. 4. El sistema le muestra al usuario el formulario con la imagen previamente cargada en el campo de imagen de producto.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecto en los campos de Nombre de Producto, Descripción, Código, Cantidad, Precio de costo, Precio de venta, Porcentaje de descuento y Tipo de producto, el sistema lo devolverá a la pantalla de Agregar producto y devolverá un mensaje de error en pantalla.
Flujo alternativo de código ya registrado	Si el usuario ingresa un código de producto que ya fue previamente ingresado en el sistema, devolverá al usuario a la pantalla de agregar producto y mostrará un error en pantalla.
Requerimientos especiales	
No aplica.	
Postcondiciones	
Una vez ejecutado el caso de uso, el sistema registra el producto en la base de datos y puede ser utilizado en los módulos de compras y facturación.	

Fuente: Elaboración propia.

Tabla 26. *Caso de uso de consulta de productos.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.18
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Consulta de productos
Autor del caso de uso:	Diego Méndez Rodríguez

Actores relacionados:	Usuario Administrador Usuario Regular
Precondiciones:	El usuario debe haber ingresado previamente con sus credenciales al sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario expande la opción del menú de navegación Productos y selecciona la opción del submenú Lista de Productos. 2. El sistema muestra un listado en forma resumida de los productos registrados en el sistema. 3. El usuario puede buscar un registro en específico dentro del listado con base en el subflujo búsqueda de producto. 4. El usuario selecciona el producto a consultar y presiona el botón Detalles de producto. 5. El sistema muestra la información detallada del producto. 	
Subflujos	
Subflujo búsqueda de producto	<ol style="list-style-type: none"> 1. Dentro de la vista de listado de productos, en el campo de búsqueda, el usuario digita ya sea el nombre de producto o código para buscar un producto. 2. El sistema muestra en pantalla el producto o productos resultantes en la búsqueda.
Flujos alternos	
Flujo alternativo de inexistencia de producto	En el caso de que el usuario haga una búsqueda de algún registro que no coincida ningún criterio con los productos existentes, el sistema devuelve a la pantalla de listado de productos y muestra un mensaje de que no existen productos relacionados.
Requerimientos especiales	
No aplica.	
Postcondiciones	
Al finalizar el caso de uso, el sistema muestra el listado completo de productos.	

Fuente: Elaboración propia.

Tabla 27. Caso de uso de modificación de producto.

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.19
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Modificación de producto
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	<p>El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema.</p> <p>El usuario tiene que ejecutar previamente el caso de uso CU.18.</p>

Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Editar dentro de la pantalla de Detalles del Producto. 2. El sistema muestra al usuario el formulario de registro de producto, pero con los datos precargados del producto a modificar. 3. El usuario realiza las modificaciones respectivas sobre el registro a modificar. 4. El usuario puede modificar la imagen del registro, con base en el subflujo de cambio de imagen. 5. En caso de que el usuario no cambie la imagen, el sistema colocara la imagen existente en base de datos. 6. El usuario presiona el botón Modificar producto. 7. El sistema devuelve al usuario a la pantalla de Detalles para que el usuario pueda observar los cambios realizados. 	
Subflujos	
Subflujo de cambio de imagen	<ol style="list-style-type: none"> 1. Dentro de la interfaz de modificar producto, el usuario presiona el botón “Seleccionar Imagen”. 2. El sistema le muestra al usuario un explorador de archivos de Windows. 3. El usuario busca la imagen a agregar en su ordenador y la selecciona con el botón abrir del explorador de archivos. 4. El sistema le muestra al usuario el formulario con la imagen previamente cargada en el campo de imagen de producto.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa información en un formato incorrecto en los campos de Nombre de producto, Descripción, Código, Cantidad, Precio de costo, Precio de venta, Porcentaje de descuento y Tipo de producto, el sistema lo devolverá a la pantalla de modificar producto y devolverá un mensaje de error en pantalla.
Flujo alternativo código ya registrado	Si el usuario ingresa un código de producto que ya fue previamente ingresada en el sistema, devolverá al usuario a la pantalla de modificar producto y mostrará un error en pantalla.
Flujo alternativo de números negativos	Si el usuario ingresa un precio menor a cero, el sistema devuelve al usuario a la pantalla de modificar usuario y muestra un error en pantalla
Requerimientos especiales	
No aplica.	
Postcondiciones	
Una vez ejecutado el caso de uso, el sistema guarda los registros modificados en base de datos, y devuelve el usuario a la pantalla de detalles de producto para revisar la información modificada.	

Fuente: Elaboración propia.

Tabla 28. *Caso de uso de eliminación de producto.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.20
Fecha de elaboración:	05/02/2023
Descripción del caso de uso:	Eliminación de producto
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador Usuario Regular
Precondiciones:	El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema. El usuario tiene que ejecutar previamente el caso de uso CU.18.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Eliminar dentro de la pantalla de Detalles de Producto. 2. El sistema muestra una ventana emergente solicitando confirmación si realmente se desea eliminar el registro. 3. El usuario confirma que desea eliminar el registro. 4. El sistema muestra un resumen del producto a borrar y vuelve a pedir una confirmación si desea eliminar el registro mostrado en pantalla. 5. El usuario confirma que desea eliminar el registro. 6. El sistema devuelve al usuario a la pantalla de listado de productos donde ya no se mostrará el registro borrado. 	
Subflujos	
Flujos alternos	
Flujo alternativo de confirmación de Borrado	Si el usuario ingreso a la opción de borrado por error y le indica al sistema que no quiere borrar el producto, el sistema no realiza ninguna acción y devuelve a la pantalla de Detalles de producto.
Requerimientos especiales	
Para cuidar la integridad referencial en la base de datos, el borrado del registro se debe realizar a nivel lógico y no físico.	
Postcondiciones	
Al finalizar la ejecución del caso de uso, el sistema realiza la modificación de la columna de habilitado por el valor de "False" para así procesar el borrado lógico del registro.	

Fuente: Elaboración propia.

Tabla 29. *Caso de uso de registro de ventas.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.21

Fecha de elaboración:	06/02/2023
Descripción del caso de uso:	Registro de ventas
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador Usuario Regular
Precondiciones:	El usuario debe haber iniciado sesión previamente en el sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario selecciona a la opción del menú de navegación Ventas, e ingresa a la opción del submenú Agregar Venta. 2. El sistema muestra la pantalla de registro de venta. 3. El usuario digita el número de cédula del cliente. 4. El sistema muestra la información del cliente con base en el número de cédula ingresado. 5. El usuario digita el código del producto a vender. 6. El sistema muestra la información de cada producto registrado, dicha información sea nombre, descripción y precio de venta. 7. En caso de que el usuario no conozca el código del producto, puede realizar una búsqueda de este con base en el subflujo búsqueda de producto. 8. El usuario digita la cantidad de productos a vender. 9. El sistema muestra el monto de descuento, impuestos y el importe de la cantidad de productos vendidos. 10. El usuario presiona el botón Agregar Productos. El usuario realiza los pasos del 5 al 10 para cada producto vendido. 11. El sistema muestra una tabla de todos los productos registrados dentro de la venta. 12. El usuario navega a la pestaña de Procesar venta. 13. El sistema muestra la vista de procesar venta en donde contiene los campos para finalizar la venta. 14. El usuario selecciona si la venta es a contado o a crédito. Con base en los subflujos venta de contado y venta a crédito. 15. El sistema muestra el descuento, impuesto e importe total de la venta. 16. El usuario presiona el botón Procesar venta. 17. El sistema genera el ticket con el resumen de productos vendido y montos. 18. Se ejecuta el subflujo umbral de producto. 	
Subflujos	
Subflujo de búsqueda de productos	<ol style="list-style-type: none"> 1. Dentro de la pantalla de agregar productos, el usuario presiona el botón buscar producto. 2. El sistema muestra la pantalla de búsqueda de productos donde contiene un formulario de búsqueda y una tabla con los registros resultantes. 3. El usuario digita el criterio de búsqueda y acciona el botón buscar. 4. El sistema actualiza la tabla de productos. 5. El usuario presiona el botón “Seleccionar” dentro del registro de producto que requiere. 6. El sistema devuelve al usuario a la pantalla de venta de producto.
Subflujo de venta de contado	<ol style="list-style-type: none"> 1. Dentro de la vista de procesar pago el usuario selecciona el <i>checkbox</i> de venta de contado.

	<p>2. El sistema muestra un campo para seleccionar el método de pago, entre efectivo, tarjeta, o transferencia.</p> <p>3. El usuario selecciona el método de pago.</p> <p>4. El sistema carga el campo de formulario de dinero recibido en caso de ser venta en efectivo, y campo de número de referencia en caso de ser tarjeta o transferencia.</p> <p>5. El usuario digita el dinero recibido en caso de venta en efectivo o el número de referencia en caso de tarjeta o transferencia.</p> <p>6. El sistema muestra el cambio que se le da al cliente en caso de ser venta en efectivo.</p>
Subflujo de venta a crédito	<p>1. Dentro de la vista de procesar pago el usuario selecciona el <i>checkbox</i> de venta a crédito.</p> <p>2. El sistema no carga ningún campo adicional y procesa los cargos al crédito de cliente finalizada la venta.</p>
Subflujo de umbral de producto	Si al realizar la venta del producto la cantidad de existencias del producto quedan por el mínimo del umbral permitido, el sistema enviará una alerta vía correo electrónico de que el producto esta pronto a acabarse.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa un formato incorrecto en cualquiera de los campos del formulario, el sistema devuelve al usuario a la pantalla de venta y muestra un error en pantalla.
Flujo alternativo de producto no existente	Si el usuario realiza la búsqueda de un producto que no fue previamente ingresado al sistema, le muestra un error al usuario indicándole que el producto no existe.
Flujo alternativo de cliente no existente	Si el usuario realiza la búsqueda de un cliente que no fue previamente ingresado al sistema, le muestra un error al usuario indicándole que el cliente no existe.
Flujo alternativo de inexistencia de productos	Si el usuario intenta vender una cantidad de productos mayor a la existente en inventario, el sistema devolverá un error en pantalla indicando que no hay existencias de ese producto para continuar la operación.
Flujo alternativo de pago incorrecto	Si el usuario en una venta a contado y en efectivo digita una cantidad de dinero recibido menor al importe total de la venta, muestra un error en pantalla indicándole que el dinero ingresado es insuficiente.
Flujo alternativo de apertura de caja	Si el usuario intenta realizar una venta sin apertura de la caja en el sistema, devuelve un error notificando que tiene que realizar la apertura de caja antes de realizar una venta.

Requerimientos especiales
El sistema tiene que generar un ticket al cliente por cada venta realizada. El sistema tiene que notificar al usuario si posterior a una venta la cantidad de productos es menor al umbral pre-definido.
Postcondiciones
Al finalizar el caso de uso el sistema registra la venta en el histórico de ventas, disminuye la cantidad de productos en el inventario y modifica las tablas respectivas de cierre de caja.

Fuente: Elaboración propia.

Tabla 30. *Caso de uso de registro de compras.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.22
Fecha de elaboración:	06/02/2023
Descripción del caso de uso:	Registro de compras
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador
Precondiciones:	El usuario debe haber ingresado previamente al sistema utilizando las credenciales de administrador de sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario selecciona a la opción del menú de navegación Compras, e ingresa a la opción del submenú Agregar Compra. 2. El sistema muestra la pantalla de registro de compra. 3. El usuario presiona el botón de búsqueda de proveedor. 4. El sistema muestra los proveedores y un formulario para buscar proveedor. 5. El usuario digita el número de cedula o nombre de proveedor en el buscador y presiona buscar. 6. El sistema muestra los proveedores coincidentes con la búsqueda. 7. El usuario selecciona el proveedor de la compra. 8. El sistema muestra la información del proveedor. 9. El usuario digita el código del producto a comprar. 10. El sistema muestra la información de cada producto registrado, dicha información sea nombre, descripción y precio de compra. 11. En caso de que el usuario no conozca el código del producto, puede realizar una búsqueda de este con base en el subflujo búsqueda de producto. 12. El usuario digita la cantidad de productos a comprar. 13. El sistema muestra el monto de descuento, impuestos y el importe de la cantidad de productos comprados. 14. El usuario presiona el botón Agregar Productos. El usuario realiza los pasos del 9 al 14 para cada producto comprado. 15. El sistema muestra una tabla de todos los productos registrados dentro de la compra. 16. El usuario navega a la pestaña de Procesar compra. 17. El sistema muestra la vista de procesar compra en donde contiene los campos para finalizar la compra. 	

<p>18. El usuario selecciona si la compra es a contado o a crédito. Con base en los subflujos compra de contado y compra a crédito.</p> <p>19. El sistema muestra el descuento, impuesto e importe total de la compra.</p> <p>20. El usuario presiona el botón Procesar compra.</p> <p>21. El sistema genera el ticket con el resumen de productos comprados y montos.</p>	
Subflujos	
Subflujo de búsqueda de productos	<ol style="list-style-type: none"> 1. Dentro de la pantalla de agregar productos, el usuario presiona el botón buscar producto. 2. El sistema muestra la pantalla de búsqueda de productos donde contiene un formulario de búsqueda y una tabla con los registros resultantes. 3. El usuario digita el criterio de búsqueda y acciona el botón buscar. 4. El sistema actualiza la tabla de productos. 5. El usuario presiona el botón “Seleccionar” dentro del registro de producto que requiere. 6. El sistema devuelve al usuario a la pantalla de compra de producto.
Subflujo de compra de contado	<ol style="list-style-type: none"> 1. Dentro de la vista de procesar pago el usuario selecciona el <i>checkbox</i> de compra de contado. 2. El sistema muestra un campo para seleccionar el método de pago, entre efectivo, tarjeta, o transferencia. 3. El usuario selecciona el método de pago. 4. El sistema carga el campo de formulario de dinero recibido en caso de ser compra en efectivo, y campo de número de referencia en caso de ser tarjeta o transferencia. 5. El usuario digita el dinero pagado en caso de venta en efectivo o el número de referencia en caso de tarjeta o transferencia.
Subflujo de compra a crédito	<ol style="list-style-type: none"> 1. Dentro de la vista de procesar pago el usuario selecciona el <i>checkbox</i> de venta a crédito. 2. El sistema no carga ningún campo adicional y procesa los cargos al crédito de cliente finalizada la venta.
Flujos alternos	
Flujo alternativo de validación	Si el usuario ingresa un formato incorrecto en cualquiera de los campos del formulario, el sistema devuelve al usuario a la pantalla de compra y muestra un error en pantalla.
Flujo alternativo de producto no existente	Si el usuario realiza la búsqueda de un producto que no fue previamente ingresado al sistema, le muestra un error al usuario indicándole que el producto no existe.
Flujo alternativo de proveedor no existente	Si el usuario realiza la búsqueda de un proveedor que no fue previamente ingresado al sistema, le muestra un error al usuario indicándole que el proveedor no existe.

Flujo alternativo de pago incorrecto	Si el usuario en una compra a contado y en efectivo digita una cantidad de dinero recibido menor al importe total de la compra, muestra un error en pantalla indicándole que el dinero ingresado es insuficiente.
Flujo alternativo de apertura de caja	Si el usuario intenta realizar una compra sin apertura de la caja en el sistema, devuelve un error notificando que tiene que realizar la apertura de caja antes de realizar una compra.
Requerimientos especiales	
El sistema tiene que solicitar el campo de número de factura del proveedor como referencia de la compra.	
Postcondiciones	
Al finalizar el caso de uso el sistema registra la compra en el histórico de compra, aumenta la cantidad de productos en el inventario y modifica las tablas respectivas de cierre de caja.	

Fuente: Elaboración propia.

Tabla 31. *Caso de uso de apertura de caja.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.23
Fecha de elaboración:	06/02/2023
Descripción del caso de uso:	Apertura de caja
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Administrador Usuario Regular
Precondiciones:	El usuario debe haber iniciado sesión previamente en el sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario navega al menú de navegación Operaciones de Caja, y selecciona la opción apertura de caja. 2. El sistema muestra la pantalla de apertura de caja en donde muestra los campos para ingresar la cantidad de dinero inicial en caja. 3. El usuario ingresa la cantidad de dinero en caja y presiona el botón de apertura de caja. 4. El sistema devuelve al usuario a la pantalla principal del sistema. 	
Subflujos	
Flujos alternos	
Flujo alternativo de Validación	Si el usuario ingresa un formato incorrecto en cualquiera de los campos del formulario, el sistema devuelve al usuario a la pantalla de apertura de caja y muestra un error en pantalla.

Flujo alternativo de Caja Abierta	Si el usuario ya había realizado la apertura de caja anteriormente, el sistema muestra un error de que ya existe una apertura de caja.
Requerimientos especiales	
El sistema tiene que colocar la fecha de apertura de caja, según la fecha actual del sistema. El sistema asigna a la apertura de caja el usuario actualmente conectado en el sistema.	
Postcondiciones	
No aplica.	

Fuente: Elaboración propia.

Tabla 32. *Caso de uso de cierre de caja.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.24
Fecha de elaboración:	07/02/2023
Descripción del caso de uso:	Cierre de caja
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Regular Usuario Administrador
Precondiciones:	El usuario debe haber iniciado sesión previamente en el sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario navega al menú de navegación Operaciones de Caja, y selecciona la opción cierre de caja. 2. El sistema muestra la pantalla de cierre de caja en donde muestra los totales de ventas diarios y los campos para ingresar la cantidad de dinero final en caja. 3. El usuario ingresa la cantidad de dinero en caja y presiona el botón de cierre de caja. 4. El sistema devuelve al usuario a la pantalla principal del sistema. 	
Subflujos	
Flujos alternos	
Flujo alternativo de Validación	Si el usuario ingresa un formato incorrecto en cualquiera de los campos del formulario, el sistema devuelve al usuario a la pantalla de cierre de caja y muestra un error en pantalla.
Flujo alternativo de Caja Cerrada	Si el usuario ya había realizado el cierre de caja anteriormente, el sistema muestra un error de que ya existe un cierre de caja y tiene que abrir nuevamente la caja.
Requerimientos especiales	
El sistema tiene que colocar la fecha de cierre de caja, según la fecha actual del sistema. El sistema asigna al cierre de caja el usuario actualmente conectado en el sistema.	

Postcondiciones
Al finalizar el caso de uso el sistema registra el cierre de caja en base de datos y puede ser consultado por el administrador del sistema.

Fuente: Elaboración propia.

Tabla 33. *Caso de uso de análisis de datos.*

Prototipo: Prototipo funcional para la gestión de ventas de la Librería y Bazar Dieka ubicada en Moravia.	
Numero de caso de uso:	CU.25
Fecha de elaboración:	07/02/2023
Descripción del caso de uso:	Análisis de datos
Autor del caso de uso:	Diego Méndez Rodríguez
Actores relacionados:	Usuario Regular Usuario Administrador
Precondiciones:	El usuario debe haber iniciado sesión previamente en el sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario navega al menú de navegación Análisis de datos. 2. El sistema muestra los distintos reportes que se pueden consultar en el sistema. 3. El usuario selecciona el reporte de análisis de datos que quiere consultar. 4. El sistema muestra el reporte de análisis de datos en pantalla con sus respectivas gráficas y tablas. 5. El usuario interactúa con las distintas gráficas presentes en el reporte. 	
Subflujos	
Flujos alternos	
Requerimientos especiales	
Postcondiciones	
Al finalizar el caso de uso el sistema muestra las distintas gráficas de análisis de datos al usuario.	

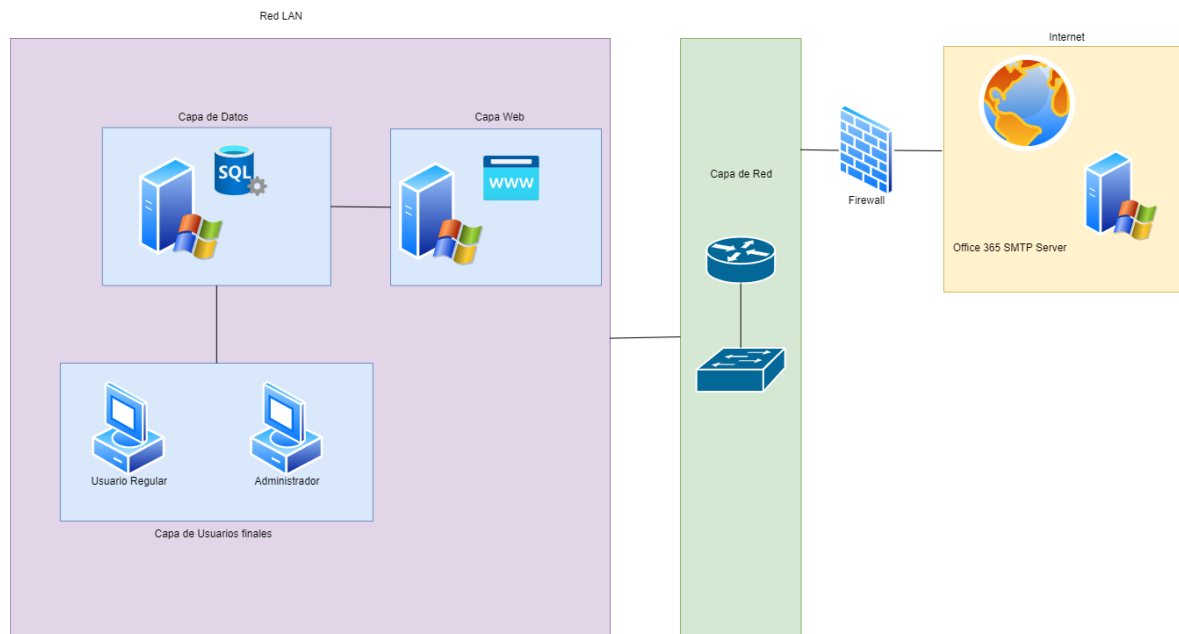
Fuente: Elaboración propia.

Arquitectura del sistema

La arquitectura del sistema es aquella en donde varios elementos de infraestructura se relacionan para cumplir una función dentro de un sistema de información. Dentro de la arquitectura del prototipo de ventas para la Librería y Bazar Dieka ubicada en Moravia, se

observa un servidor web con la aplicación IIS Express instalada para alojar la aplicación, adicional un servidor de base de datos con SQL Standard Edition. Debido a la poca transaccionalidad del sistema, pues no se ocupa de gran cantidad de recursos y procesamiento, ambos servidores se alojarán en una computadora con buenos recursos que ya cuenta actualmente el cliente. El equipo donde está alojada la aplicación tiene que contar con salida a Internet para el envío de correos utilizando el servidor SMTP de Office 365.

Figura 2. Diagrama de arquitectura del sistema.

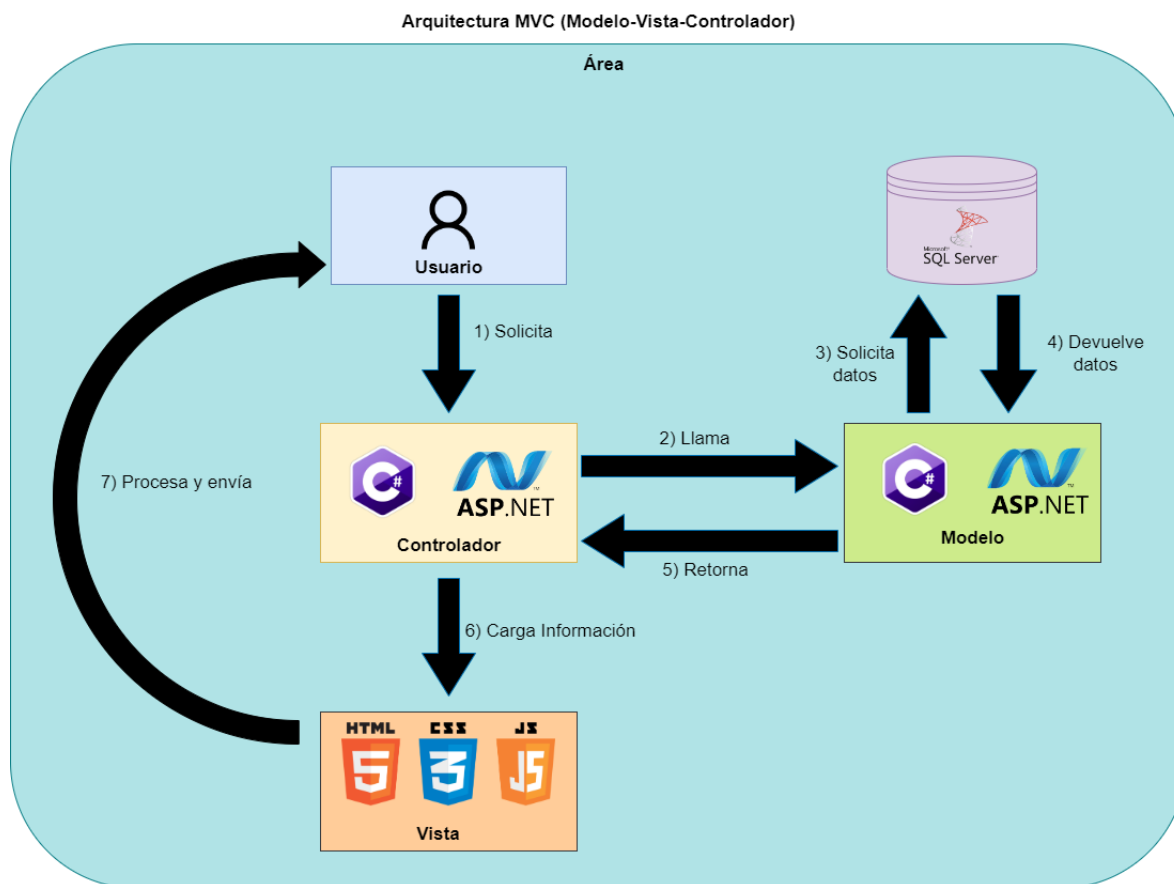


Fuente: Elaboración propia.

Arquitectura del software

Para el presente prototipo, se diseña utilizando la arquitectura de modelo vista y controlador, en donde la capa de vistas es donde se diseñan las interfaces de la aplicación y los scripts que se ejecutan del lado del cliente en esta capa se utilizan tecnologías como HTML, CSS y JavaScript, en el caso de la capa controlador se trabaja la parte interna de la aplicación en la que se realizan todos los cálculos necesarios de la información antes de ser almacenada en la base de datos, en esta capa se utilizan las tecnologías C# y ASP.NET CORE MVC. Por último, la capa de modelo es aquella donde se almacena la información procesada por la capa lógica, en esta se utiliza la tecnología de Microsoft SQL Server.

Figura 3. Diagrama de arquitectura del software.



Fuente: Elaboración propia.

Diseño de interfaces

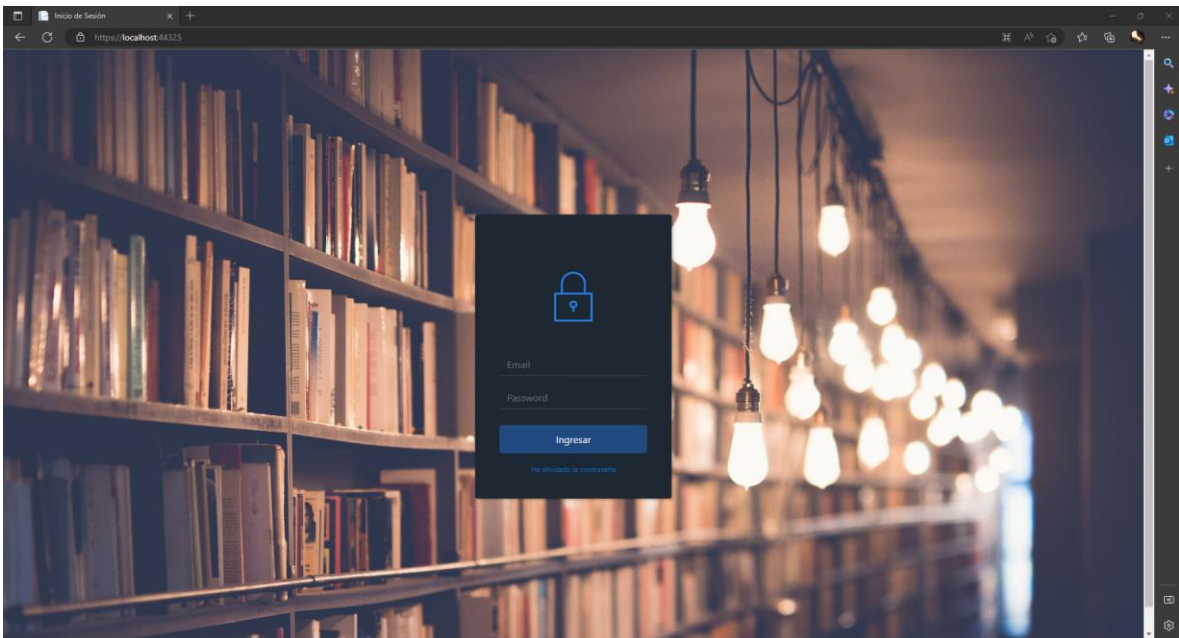
Según lo observado en el apartado anterior, la capa de presentación será desarrollada utilizando lenguaje HTML, con personalización de estilos en CSS, utilizando el *framework* de Bootstrap. Por lo tanto, todas las interfaces del prototipo serán gráficas. Dichas interfaces se diseñaron para que cumplan las siguientes características.

1. Amigables con el usuario: las interfaces diseñadas buscan que la experiencia de usuario sea satisfactoria, al momento de utilizar el sistema, aplicando características como poca cantidad de controles por pantalla, tamaño apropiado de los campos y de las fuentes.

2. Interfaces estandarizadas: se buscó, dentro del modelado de interfaces, un diseño estandarizado donde al usuario le sea sencillo utilizar las interfaces, ya que tendrán la misma estructura o estructuras muy similares.
3. Últimas tecnologías: las interfaces fueron desarrolladas en las últimas versiones de HTML y Bootstrap, asegurando así que el sitio cuente con las últimas funcionalidades publicadas en el mercado.

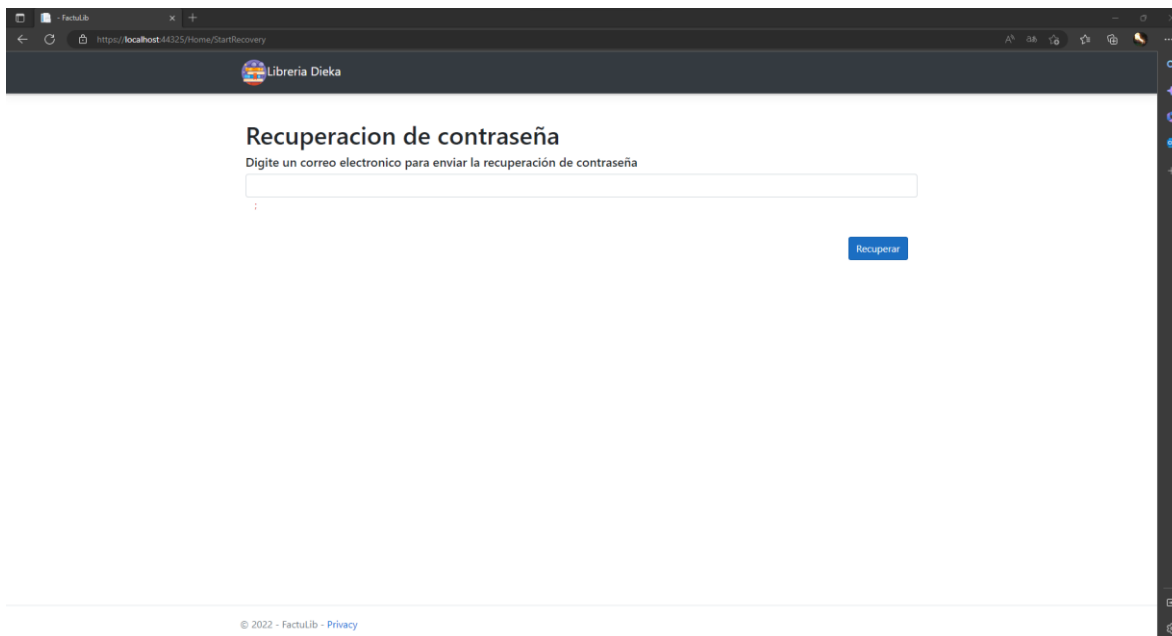
Las siguientes pantallas muestran todas las interfaces que se encontraron en el prototipo:

Figura 4. *Pantalla de inicio de sesión.*



Fuente Elaboración propia.

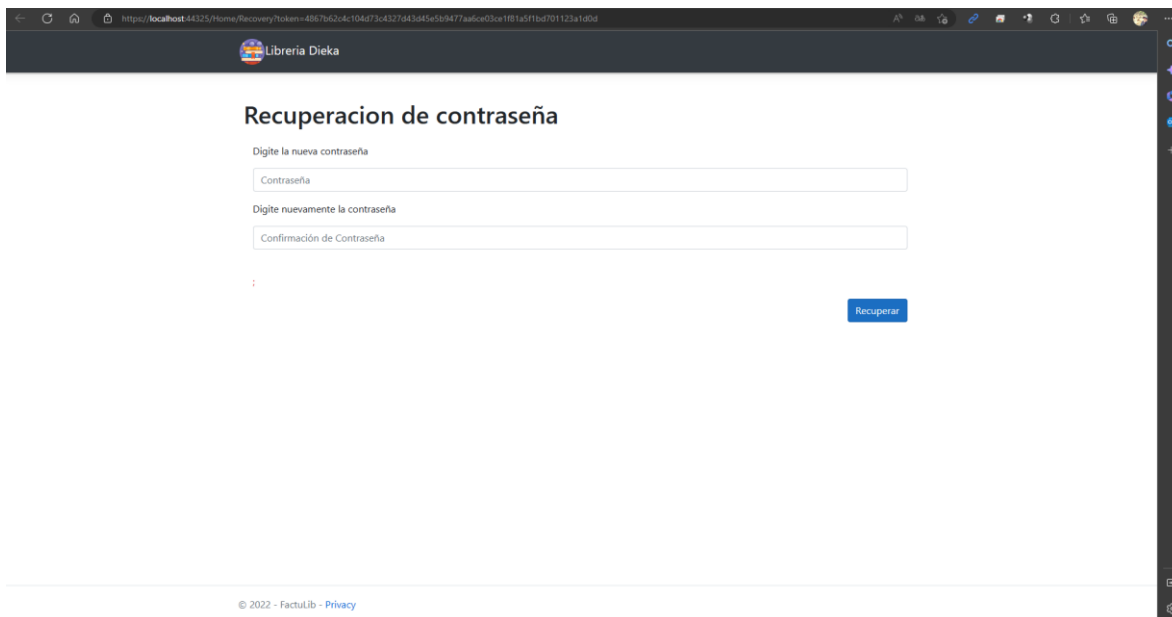
Figura 5. *Pantalla de recuperación de contraseña.*



The screenshot shows a web browser window with the URL `https://localhost:44325/Home/StartRecovery`. The page header features the 'Librería Dieka' logo. The main heading is 'Recuperacion de contraseña'. Below it, the instruction 'Digite un correo electronico para enviar la recuperacion de contraseña' is followed by a single text input field. A blue 'Recuperar' button is positioned to the right of the input field. At the bottom of the page, there is a footer with the text '© 2022 - FactuLib - Privacy'.

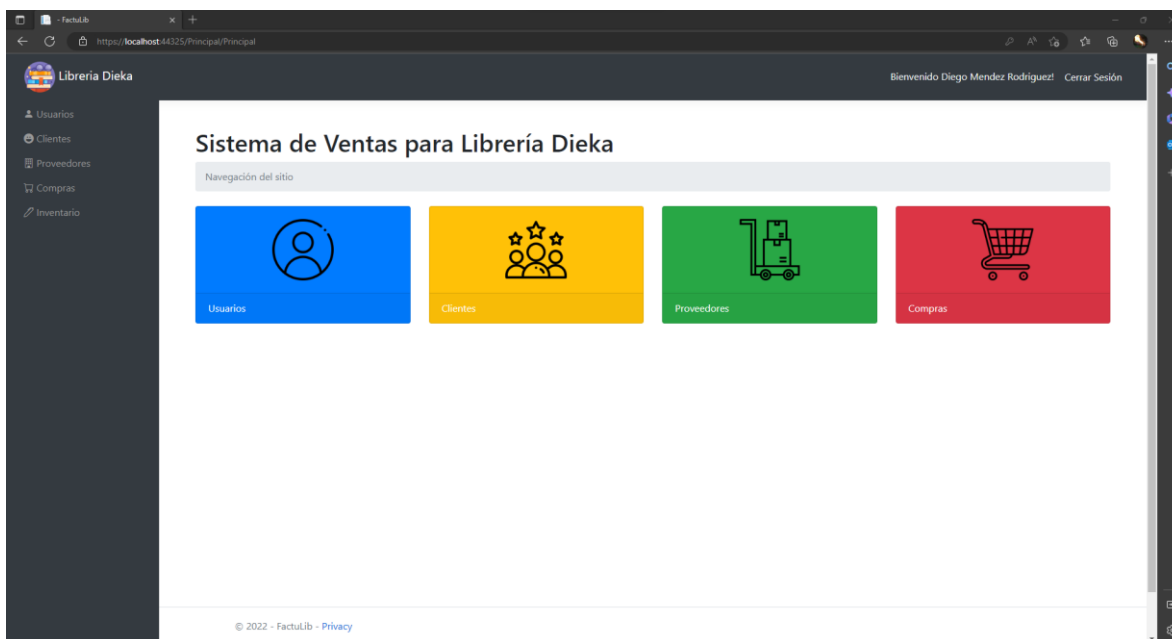
Fuente: Elaboración propia.

Figura 6. *Pantalla de cambio de contraseña.*

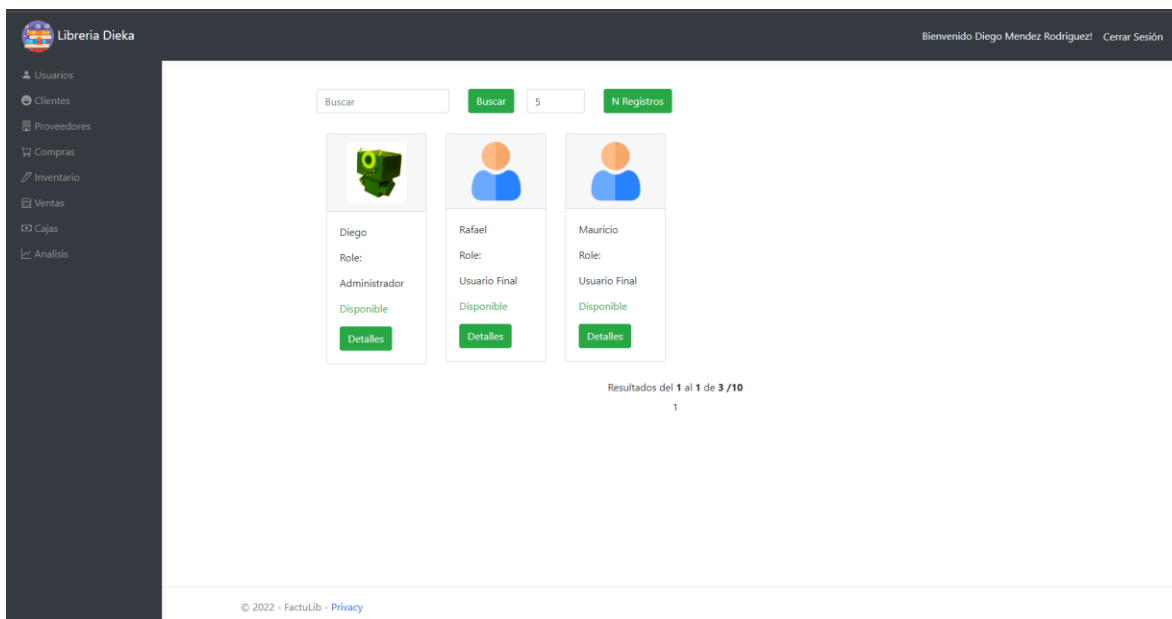


The screenshot shows a web browser window with the URL `https://localhost:44325/Home/RecoveryToken=8967b624c104d73c4327d43d45e509477aafce03e1f81d5f1bd701123a10bd`. The page header features the 'Librería Dieka' logo. The main heading is 'Recuperacion de contraseña'. Below it, the instruction 'Digite la nueva contraseña' is followed by a text input field labeled 'Contraseña'. Below that, the instruction 'Digite nuevamente la contraseña' is followed by a text input field labeled 'Confirmación de Contraseña'. A blue 'Recuperar' button is positioned to the right of the second input field. At the bottom of the page, there is a footer with the text '© 2022 - FactuLib - Privacy'.

Fuente: Elaboración propia.

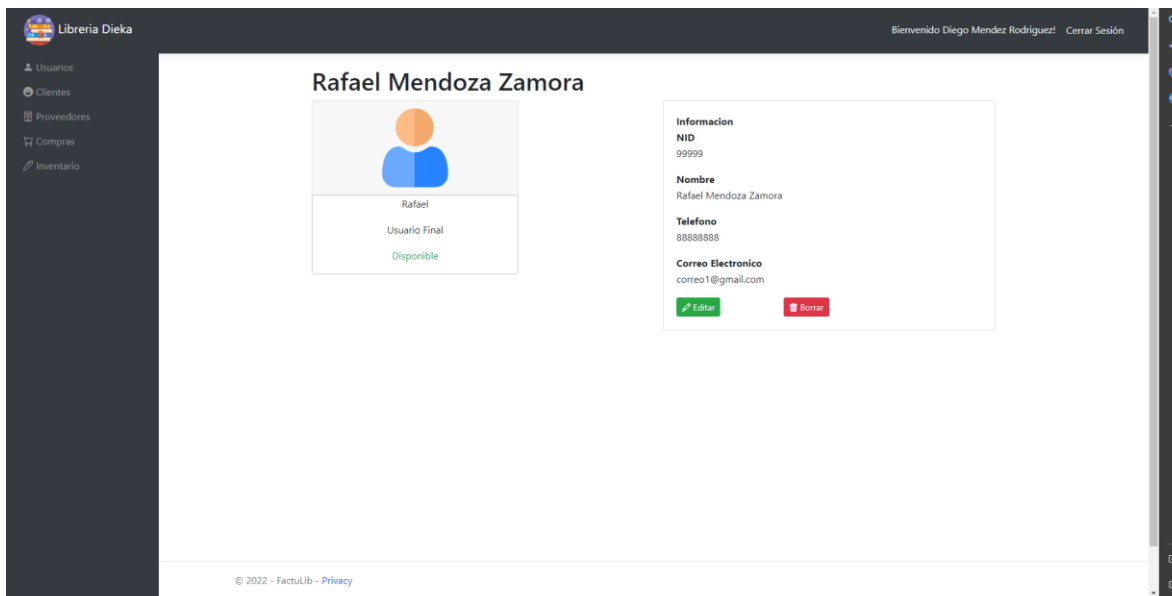
Figura 7. *Pantalla principal.*

Fuente: Elaboración propia.

Figura 8. *Pantalla de lista de usuarios.*

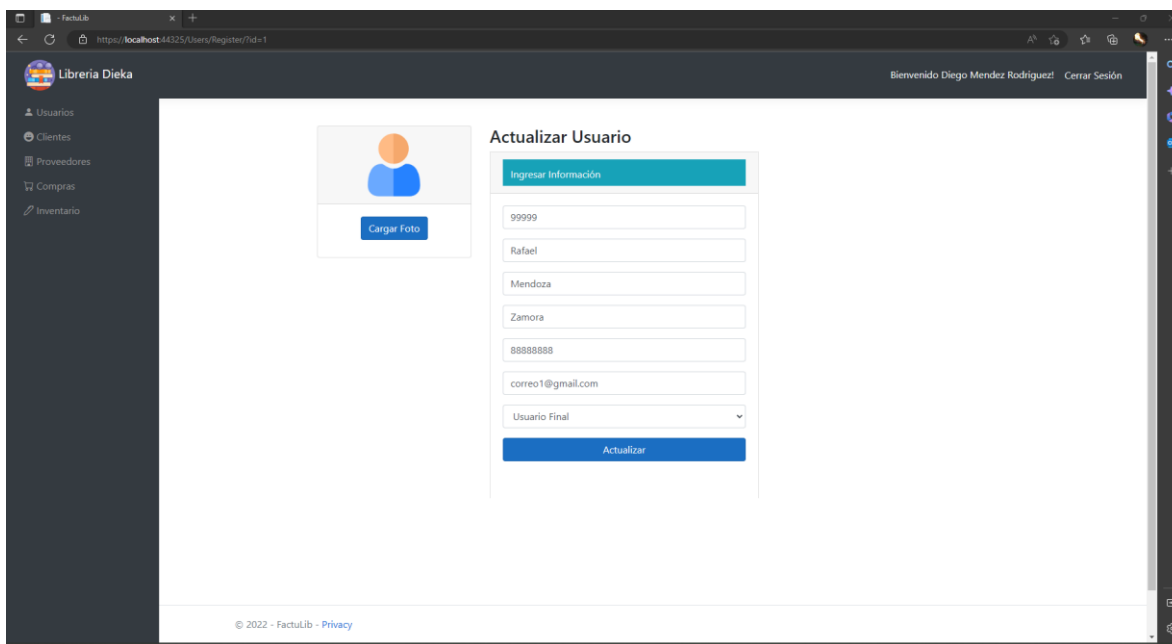
Fuente: Elaboración propia

Figura 9. Pantalla de detalles de usuario.



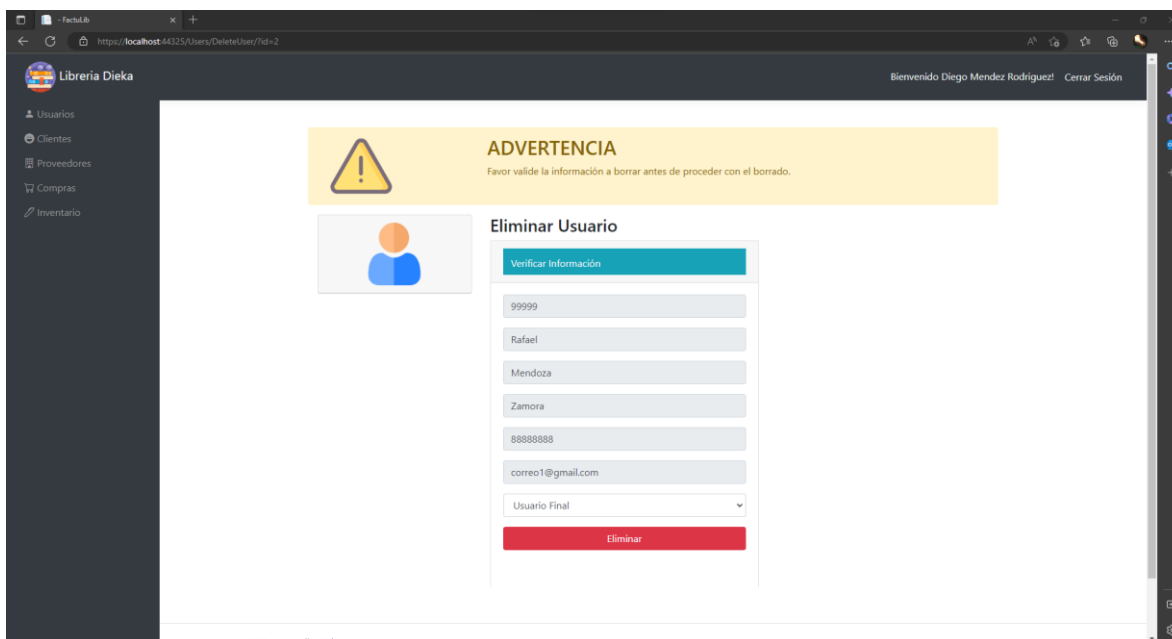
Fuente: Elaboración propia.

Figura 10. Pantalla de actualizar usuario.



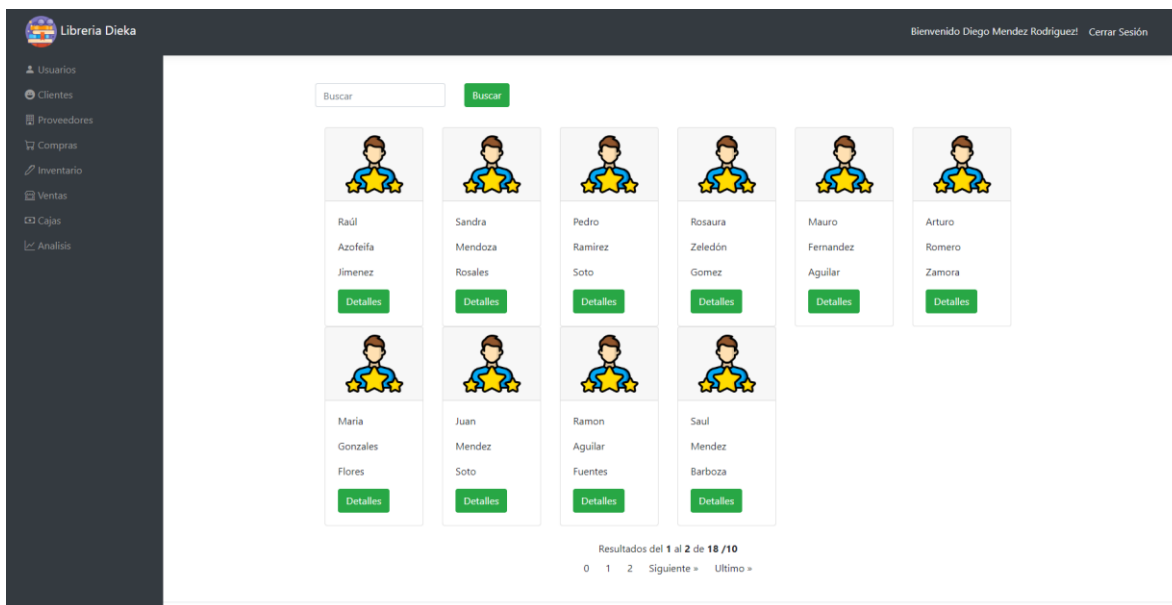
Fuente: Elaboración propia.

Figura 11. Pantalla de borrado de usuario.



Fuente: Elaboración propia.

Figura 12. Pantalla de lista de clientes.



Fuente: Elaboración propia.

Figura 13. Pantalla de registro de cliente

Librería Dieka

Bienvenido Diego Mendez Rodriguez! Cerrar Sesión

Registrar Cliente

Ingresar Información

--Seleccione tipo de identificación--

Nombre

Primer Apellido

Segundo Apellido

Numero de Telefono

Correo Electronico

Direccion

--Seleccionar Provincia--

--Seleccionar Canton--

--Seleccionar Distrito--

¿Elegible Credito?

Registrar

Fuente: Elaboración propia.

Figura 14. Pantalla de Detalles de cliente.

Librería Dieka

Bienvenido Diego Mendez Rodriguez! Cerrar Sesión

Raúl Azofeifa Jimenez

Raúl

Credito: Disponible

Credito de Cliente

Información

Numero de Cédula
999999999

Nombre
Raúl Azofeifa Jimenez

Telefono
88888888

Correo Electronico
correo9000@gmail.com

Provincia
Atajuela

Canton
San Carlos

Distrito
La Palmera

Direccion Exacta
Casa Esquinera Blanca

Editar Borrar

© 2022 - FactuLib - Privacy

Fuente: Elaboración propia.

Figura 15. Pantalla de actualizar cliente.

Librería Dieka

Bienvenido Diego Mendez Rodriguez! Cerrar Sesión

Usuarios
Clientes
Proveedores
Compras
Inventario

Actualizar Cliente

Ingresar Información

123412341

Mauros

Fernandez

Aguilar

88888888

correo91@gmail.com

Casa Azul

--Seleccionar Provincia--

--Seleccionar Canton--

--Seleccionar Distrito--

¿Elegible Credito?

Actualizar

Fuente: Elaboración propia.

Figura 16. Pantalla de borrado de cliente.

Librería Dieka

Bienvenido Diego Mendez Rodriguez! Cerrar Sesión

Usuarios
Clientes
Proveedores
Compras
Inventario

ADVERTENCIA

Favor valide la información a borrar antes de proceder con el borrado.

Eliminar Cliente

Verificar Información

123412341

Mauro

Fernandez

Aguilar

88888888

correo91@gmail.com

Casa Azul

¿Elegible Credito?

Eliminar

Fuente: Elaboración propia.

Figura 17. Pantalla de crédito de clientes.

Reportes de pago

Cuotas
Deuda: €800 000,00
Pago: €0 000,00
Cuotas por mes: €50 000,00
Fecha sigt. pago: 03/feb./2023
Ticket: 0000000004

Cuotas

Efectuar pago

Historico de Pagos

Deuda Inicial	Deuda Actual	Pago Recibido	Fecha	Opciones
€1 000 000,00	€800 000,00	€60 000,00	3/1/2023	Reporte
€1 000 000,00	€850 000,00	€100 000,00	28/12/2022	Reporte
€1 000 000,00	€900 000,00	€50 000,00	28/12/2022	Reporte
€1 000 000,00	€950 000,00	€60 000,00	28/12/2022	Reporte


© 2022 - FactuLib - Privacy

Fuente: Elaboración propia.


Figura 18. Pantalla de listado de proveedores.

Proveedores - FactuLib


Buscar **Buscar**



9999999999
Studmark
Detalles



8888888888
Jimenes y Tanzi2
Detalles



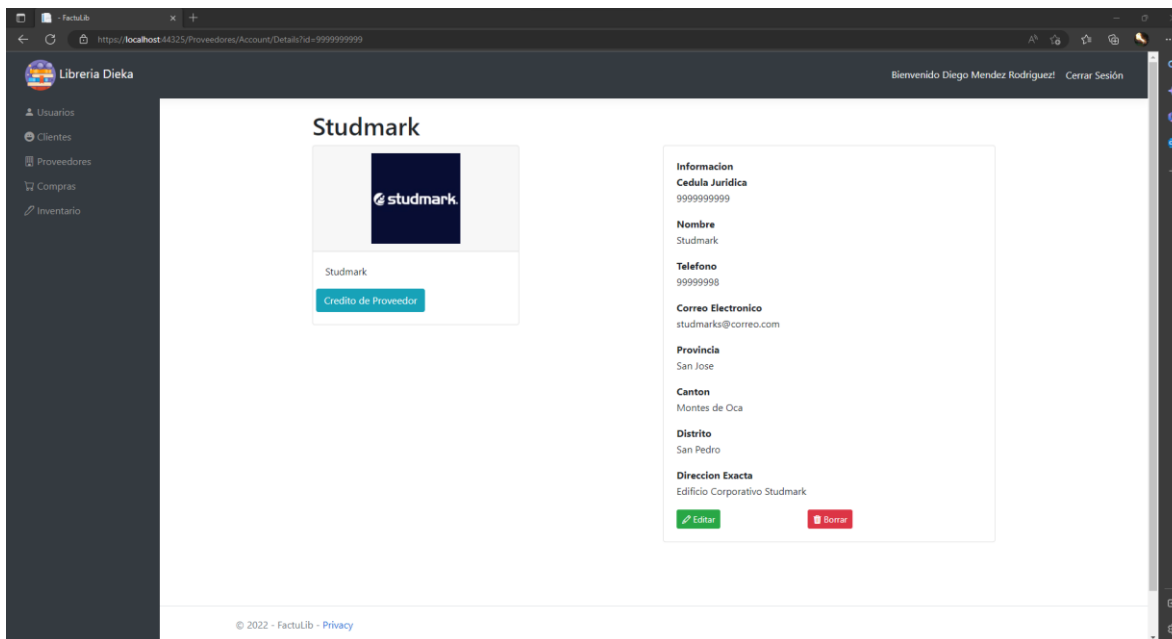
5698569854
Proveedor4
Detalles

Resultados del 1 al 1 de 3 /10

© 2022 - FactuLib - Privacy

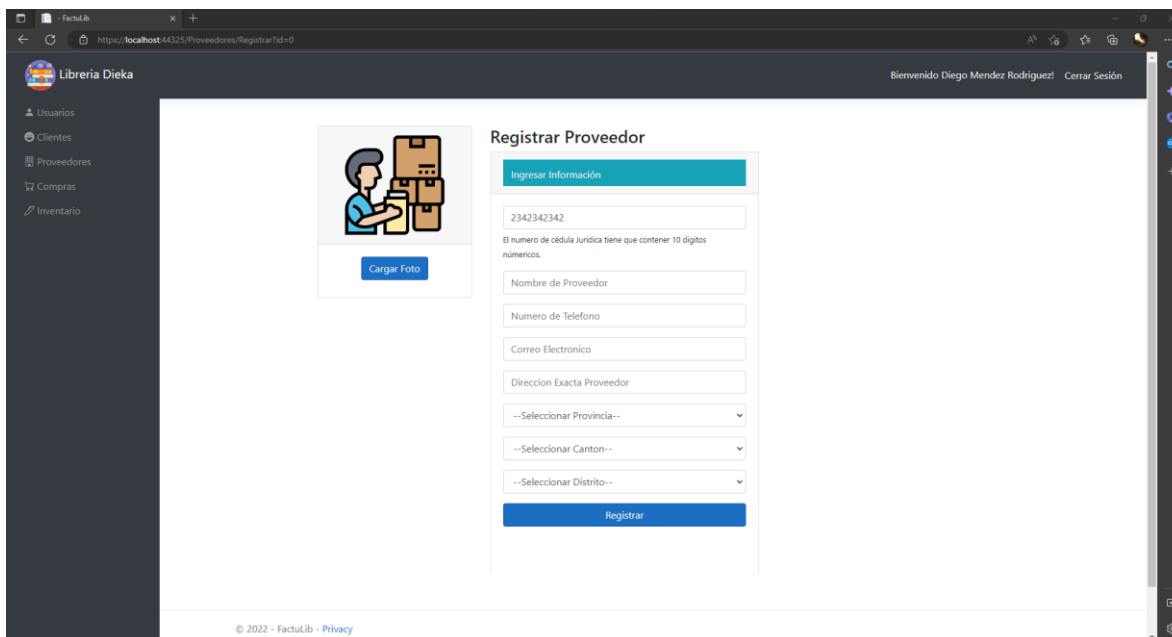
Fuente: Elaboración propia.

Figura 19. Pantalla de detalles de proveedor.



Fuente: Elaboración propia.

Figura 20. Pantalla de registro de proveedor.



Fuente: Elaboración propia.

Figura 21. Pantalla de actualizar proveedor.

Librería Dieka

Bienvenido Diego Mendez Rodriguez! Cerrar Sesión

Usuarios
Clientes
Proveedores
Compras
Inventario

Actualizar Proveedor

Ingresar Información

9999999999

El número de cédula Jurídica tiene que contener 10 dígitos numéricos.

Studmark

99999998

studmarks@correo.com

Edificio Corporativo Studmark

--Seleccionar Provincia--

--Seleccionar Cantón--

--Seleccionar Distrito--

Actualizar

© 2022 - FactuLib - Privacy

Fuente: Elaboración propia.

Figura 22. Pantalla de borrado de proveedor.

Librería Dieka

Bienvenido Diego Mendez Rodriguez! Cerrar Sesión

Usuarios
Clientes
Proveedores
Compras
Inventario

Eliminar Proveedor

ADVERTENCIA
Favor valide la información a borrar antes de proceder con el borrado.

Verificar Información

9999999999

Studmark

99999998

studmarks@correo.com

Edificio Corporativo Studmark

Eliminar

© 2022 - FactuLib - Privacy

Fuente: Elaboración propia.

Figura 23. Pantalla de crédito de proveedor.

The screenshot displays the 'Credito de Proveedor' (Supplier Credit) interface. On the left, a summary card shows the following details:

- Cuotas:** Deuda: 1 219 135,00; Pago: 60 000,00; Cuotas por mes: 50 000,00; Fecha del Pago: 01/feb./2023; Ticket: 0000000006.
- Cuotas:** A section with a 'Pago' input field and an 'Efectuar pago' button.

On the right, a table lists the payment installments with columns for 'Deuda Inicial', 'Deuda Actual', 'Pago Recibido', 'Fecha', and 'Opciones'. The table data is as follows:

Deuda Inicial	Deuda Actual	Pago Recibido	Fecha	Opciones
€1 269 135,00	€1 219 135,00	€60 000,00	1/2/2023	Reporte
€852 955,00	€802 955,00	€50 000,00	29/1/2023	Reporte
€872 840,00	€822 840,00	€60 000,00	29/1/2023	Reporte
€1 000 000,00	€850 000,00	€60 000,00	11/1/2023	Reporte
€1 000 000,00	€900 000,00	€50 000,00	6/1/2023	Reporte
€1 000 000,00	€950 000,00	€60 000,00	6/1/2023	Reporte

Fuente: Elaboración propia.

Figura 24. Pantalla de registro de productos, compra.

The screenshot displays the 'Registrar Compra' (Record Purchase) interface. On the left, a form for recording a purchase includes the following fields:

- Proveedor:** Search field.
- Id.Prod:** Input field with value '0'.
- Nombre de Producto:** Search field.
- Descripción:** Input field.
- Cant:** Input field with value '0'.
- Precio:** Input field with value '€ 0'.
- Descuento:** Input field with value '% 0'.

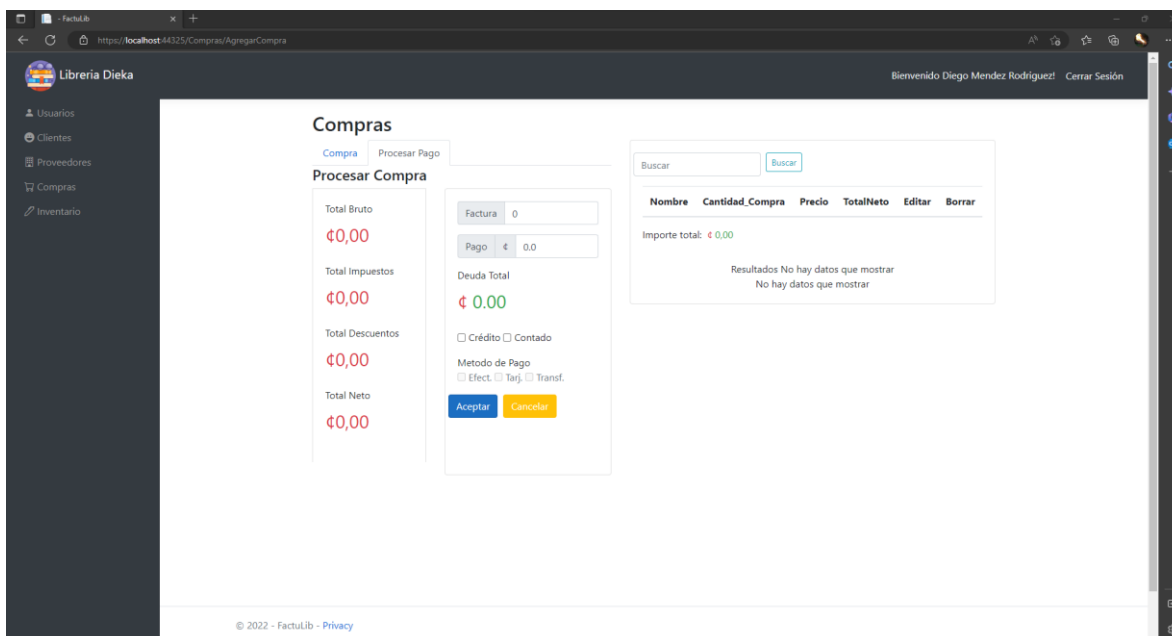
Summary of totals:

- T.Bruto: € 0.00
- Desc: € 0.00
- IVA: € 0.00
- T.NETO: € 0.00

On the right, a search table is visible with columns: 'Nombre', 'Cantidad Compra', 'Precio', 'Total Neto', 'Editar', and 'Borrar'. The table shows 'Importe total: € 0.00' and a message: 'Resultados No hay datos que mostrar'.

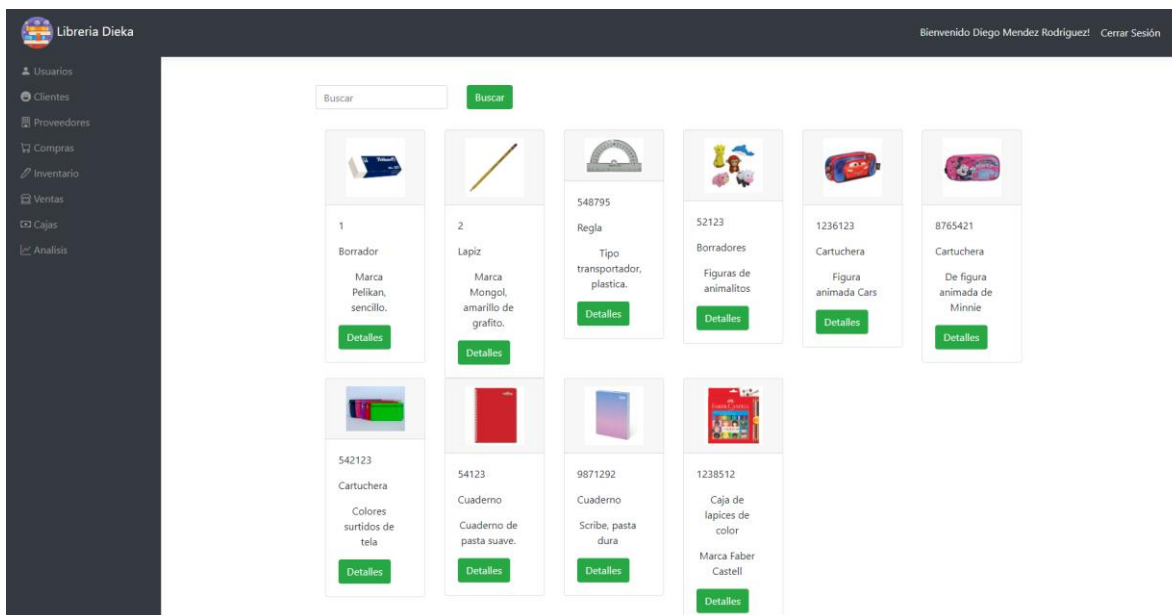
Fuente: Elaboración propia.

Figura 25. Pantalla de procesamiento de compra.



Fuente: Elaboración propia.

Figura 26. Pantalla de listado de productos.



Fuente: Elaboración propia.

Figura 27. Pantalla de registro de producto.

The screenshot shows the 'Registrar Producto' page in the 'Librería Dieka' application. The page has a dark sidebar on the left with navigation options: Usuarios, Clientes, Proveedores, Compras, and Inventario. The main content area is titled 'Registrar Producto' and contains a form with the following fields:

- Ingresar Información** (button)
-
-
-
-
-
-
-
-
-

At the bottom of the page, there is a copyright notice: © 2022 - FactuLib - Privacy.

Fuente: Elaboración propia.

Figura 28. Pantalla de detalle de producto.

The screenshot shows the 'Borradores' page in the 'Librería Dieka' application. The page has a dark sidebar on the left with navigation options: Usuarios, Clientes, Proveedores, Compras, Inventario, Ventas, Cajas, and Análisis. The main content area is titled 'Borradores' and contains a product card and a detailed information panel.

Product Card:

-
- Figuras de animalitos**
-

Information Panel:

- Información**
- Código de Producto:** 52123
- Nombre de Producto:** Borradores
- Cantidad de Producto:** 78
- Precio de Costo:** 5000,00
- Porcentaje Descuento:** %0
- Tipo de Producto:** Librería
-
-

At the bottom of the page, there is a copyright notice: © 2022 - FactuLib - Privacy.

Fuente: Elaboración propia.

Figura 29. *Interfaz de Apertura de caja*

Librería Dieka Bienvenido Diego Mendez Rodriguez! Cerrar Sesión

Usuarios Clientes Proveedores Compras Inventario Ventas Cajas Analisis

Apertura de Caja

Digite la cantidad de monedas o billetes con los que se abre la caja.

Monedas	Billetes	Totales
€5 0	€1000 0	Total Dinero en caja: € 0.00 Total Dinero en cuentas: € 0.00
€10 0	€2000 0	
€25 0	€5000 0	
€50 0	€10000 0	
€100 0	€20000 0	
€500 0	€50000 0	
Cuentas 0		

Aperturar Caja

© 2022 - FactuLib - Privacy

Fuente: Elaboración propia.

Figura 30. *Interfaz de Cierre de caja*

Librería Dieka Bienvenido Diego Mendez Rodriguez! Cerrar Sesión

Usuarios Clientes Proveedores Compras Inventario Ventas Cajas Analisis

Cierre de Caja

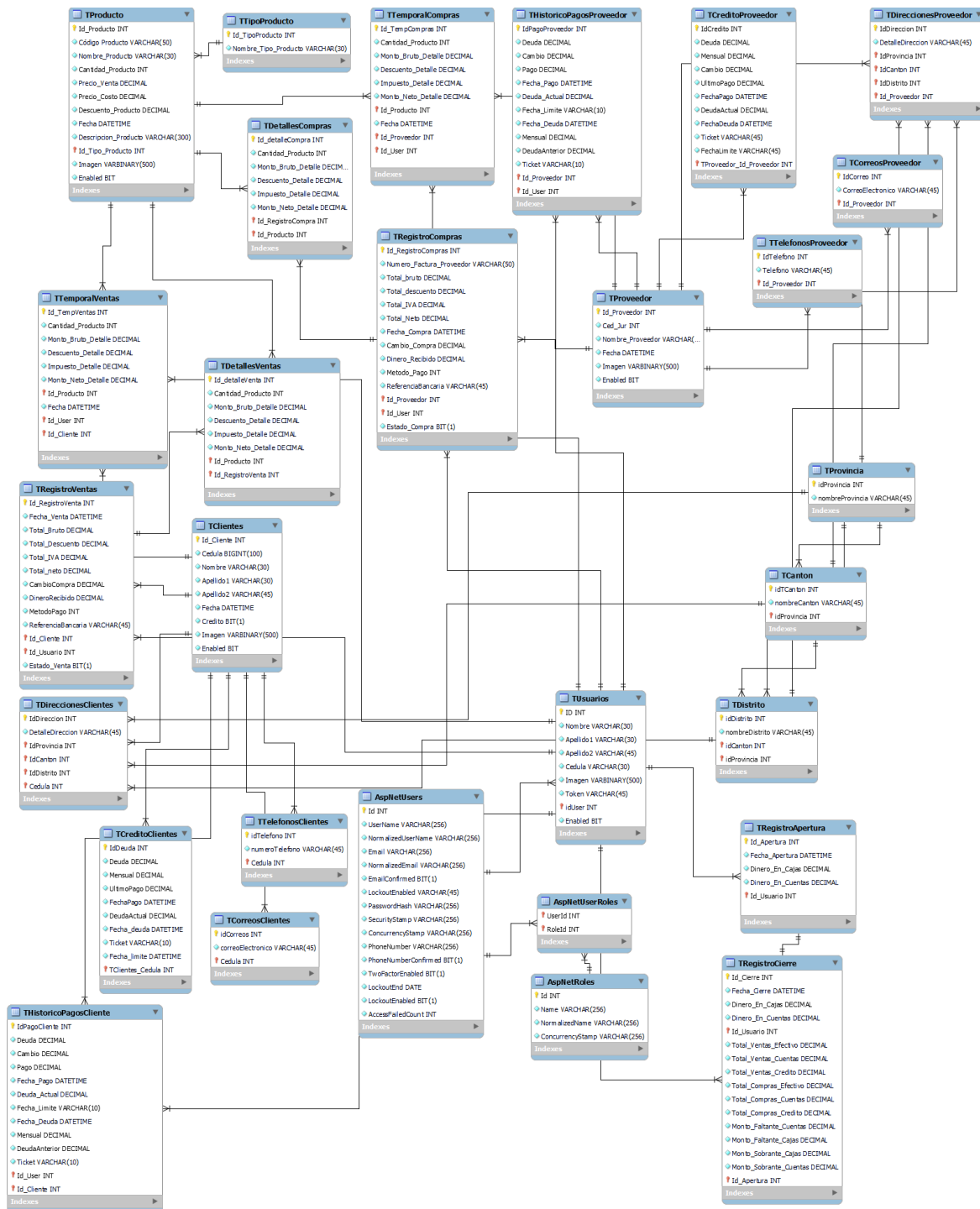
Digite la cantidad de monedas o billetes con los que se cierra la caja.

Monedas	Billetes	Totales
€5 0	€1000 0	Total Dinero en caja: € 0.00 Total Dinero en cuentas: € 0.00 Total Ventas Efectivo: € 40,00 Total Ventas Cuentas: € 40,00 Total Ventas Credito: € 40,00 Total Compras Efectivo: € 115 000,00 Total Compras Credito: € 40,00 Total Compras Cuentas: € 40,00 Total Caja Apertura: € 40,00 Total Cuentas Apertura: € 40,00
€10 0	€2000 0	
€25 0	€5000 0	
€50 0	€10000 0	
€100 0	€20000 0	
€500 0	€50000 0	
Cuentas 0		

Cerrar Caja

Diseño de Base de datos

Figura 31. Diagrama entidad - relación de base de datos



Fuente: Elaboración propia.

Diccionario de datos

Tabla 34. Diccionario de datos, TUsers.

Nombre de Tabla		TUsers
Descripción		
Esta tabla se utiliza para almacenar información de detalles específicos que va a contener un usuario que ingrese y utilice el sistema.		
Nombre de Columna	Tipo de Dato	Descripción
ID	Nvarchar	Llave primaria es un atributo incremental que identifica el usuario dentro de la tabla.
Nombre	Nvarchar	Nombre de la persona que será el propietario del usuario.
Apellido 1	Nvarchar	Primer Apellido de la persona que será el propietario del usuario.
Apellido 2	Nvarchar	Segundo Apellido de la persona que será propietario del usuario.
Cedula	Int	Cedula de la persona que será el propietario del usuario.
Email	Nvarchar	Correo electrónico de la persona la cual será propietaria del usuario.
idUser	Nvarchar	Este campo contiene el ID usuario en relación con la tabla nativa de usuarios de ASP NET MVC.
Imagen	Varbinary	Este campo contiene la definición binaria de la foto o avatar asignado al usuario.
Enabled	BIT	Este campo define si el usuario fue borrado lógicamente de la base de datos.
Índices / Constraints		
Llave Primaria		
Nombre de Llave	Columnas	
PK_ID	ID	
Llaves foráneas		

Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_idUser	idUser	AspNetUsers	Id	Relación con la tabla nativa de ASP NET Usuarios.
Otros índices				
Nombre índice	Columnas	Descripción		

Fuente: Elaboración propia.

Tabla 35. Diccionario de datos, *AspNetUsers*.

Nombre de Tabla		AspNetUsers
Descripción		
Esta es la tabla nativa de usuarios del lenguaje ASP Net para las clases de Identity User, en esta tabla se alojan los campos de contraseña de la tabla TUser.		
Nombre de Columna	Tipo de Dato	Descripción
Id	Nvarchar(450)	Llave primaria de la tabla. Identifica el registro de usuario dentro de la tabla
UserName	Nvarchar(256)	Campo para alojar el nombre de usuario para ingresar al sistema.
NormalizedUserName	Nvarchar(256)	Campo para alojar el nombre de usuario para ingresar al sistema.
Email	Nvarchar(256)	Campo para asignar un correo electrónico al usuario.
NormalizedEmail	Nvarchar(256)	Campo para asignar un correo electrónico al usuario.
EmailConfirmed	Bit	Campo que identifica si el correo fue confirmado previamente por el usuario.
PasswordHash	Nvarchar(Max)	Campo para alojar el hash de la contraseña del usuario.
SecurityStamp	Nvarchar(Max)	Security Stamp relacionado con el hash de la contraseña.
ConcurrencyStamp	Nvarchar(Max)	Concurrency stamp relacionado con el hash de la contraseña.

PhoneNumber	Nvarchar(Max)	Número de teléfono del usuario.		
PhoneNumberConfirmed	BIT	Campo que identifica si el número de teléfono fue confirmado previamente por el usuario.		
TwoFactorsEnabled	BIT	Campo que identifica si el usuario tiene doble factor de autenticación activado.		
LockOutEnd	DATETIME	Campo que identifica la fecha de expiración de bloqueo del usuario.		
LockOutEnabled	BIT	Campo que identifica si el usuario está bloqueado actualmente.		
AccessFailedCount	INT	Conteo de ingresos fallidos del usuario.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_ID		Id		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 36. Diccionario de datos, *AspNetUsersRoles*.

Nombre de Tabla		AspNetUsersRoles	
Descripción			
Tabla intermedia nativa del lenguaje ASP Net que relaciona los usuarios de la tabla Asp Net Users con la tabla Asp Net Roles.			
Nombre de Columna		Tipo de Dato	Descripción
User_Id		Nvarchar(450)	Campo de llave foránea que relaciona con la tabla AspNetUsers.
Role_Id		Nvarchar(450)	Campo de llave foránea que relaciona con la

		tabla AspNetRoles.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_ID		User_Id, Role_Id		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_AspNetUserRoles_AspNetUsers_UserId	User_Id	AspNetUsers	Id	Relación con la tabla nativa de ASP NET Usuarios.
FK_AspNetUserRoles_AspNetRoles_RoleId	Role_Id	AspNetRoles	Id	Relación con la tabla nativa ASP NET Roles.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 37. Diccionario de datos, *AspNetRoles*.

Nombre de Tabla		AspNetRoles	
Descripción			
Tabla nativa del lenguaje ASP NET que almacena los roles que se relacionan para un determinado usuario.			
Nombre de Columna	Tipo de Dato	Descripción	
ID	Nvarchar (450)	Llave primaria que identifica el registro de rol dentro de la tabla.	
Name	Nvarchar(256)	Nombre del rol.	
NormalizedName	Nvarchar(256)	Nombre normalizado del rol se utiliza para fines internos de la clase.	
Índices / Constraints			
Llave Primaria			
Nombre de Llave	Columnas		
PK_ID	ID		
Llaves foráneas			

Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
Otros índices				
Nombre índice	Columnas	Descripción		

Fuente: Elaboración propia.

Tabla 38. *Diccionario de datos, TProvincia.*

Nombre de Tabla		TProvincia		
Descripción				
Tabla que almacena las provincias que son consultadas por los campos de dirección de las distintas entidades.				
Nombre de Columna	Tipo de Dato	Descripción		
IdProvincia	Int	Llave primaria auto incremental que identifica el registro de provincia dentro de la tabla.		
NombreProvinciaL	Nvarchar(256)	Nombre de la provincia		
Índices / Constraints				
Llave Primaria				
Nombre de Llave	Columnas			
PK_IdProvincia	idProvincia			
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
Otros índices				
Nombre índice	Columnas	Descripción		

Fuente: Elaboración propia.

Tabla 39. *Diccionario de datos, TCantón.*

Nombre de Tabla		TCantón		
Descripción				
Tabla que almacena los cantones que están relacionados en los campos de dirección de las distintas entidades.				
Nombre de Columna	Tipo de Dato	Descripción		
IdCantón	Int	Llave primaria auto incremental que identifica el registro de cantón dentro de la tabla.		
NombreCanton	Nvarchar(256)	Nombre del cantón.		

IdProvincia	Int	Campo de llave foránea que hace relación a la tabla TProvincia.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_IdCanton		IdCanton		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TCanton_TProvincia_idProvincia	IdProvincia	TProvincia	IdProvincia	Relación foránea a la tabla de TProvincia.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 40. Diccionario de datos, TDistrito.

Nombre de Tabla		TDistrito		
Descripción				
Tabla que almacena los distritos que están relacionados en los campos de dirección de las distintas entidades.				
Nombre de Columna	Tipo de Dato		Descripción	
IdDistrito	Int		Llave primaria auto incremental que identifica el registro de distrito dentro de la tabla.	
NombreDistrito	Nvarchar(256)		Nombre del Distrito.	
IdCantón	Int		Campo de llave foránea que hace relación con la tabla TCantón.	
IdProvincia	Int		Campo de llave foránea que hace relación a la tabla TProvincia.	
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_IdDistrito		IdDistrito		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción

FK_TDistrito_ TCanton_idCanton	IdCantón	TCantón	IdCantón	Relación foránea a la tabla de TCanton.
FK_TDistrito_ TProvincia_idProvincia	IdProvincia	TProvincia	IdProvincia	Relación foránea a la tabla de TProvincia.
Otros índices				
Nombre índice		Columnas		Descripción

Fuente: Elaboración propia.

Tabla 41. *Diccionario de datos, TClientes.*

Nombre de Tabla		TClientes
Descripción		
Esta tabla se utiliza para almacenar información básica de los clientes.		
Nombre de Columna	Tipo de Dato	Descripción
Id_Cliente	Int	Llave primara de la tabla, representa un número auto incremental que identifica el registro en la tabla.
Cedula	BigInt	Campo de número de cédula, es un campo único dentro de la tabla.
Nombre	Nvarchar(256)	Nombre del cliente registrado en el sistema.
Apellido 1	Nvarchar(256)	Primer apellido del cliente registrado en el sistema.
Apellido 2	Nvarchar(256)	Segundo apellido del cliente registrado en el sistema.
Fecha	DateTime	Fecha de inclusión del cliente al sistema.
Crédito	Bit	Identifica si el cliente es elegible para otorgarle crédito en el sistema o no.
Imagen	Varbinary(Max)	Este campo contiene la definición binaria de la foto o avatar asignado al usuario.
Enabled	Bit	Este campo define si el cliente fue borrado lógicamente de la base de datos.
Índices / Constraints		

Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_Cliente		Id_Cliente		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
Otros índices				
Nombre índice		Columnas		Descripción

Fuente: Elaboración propia.

Tabla 42. Diccionario de datos, *TDirecciónCliente*.

Nombre de Tabla		TDirecciónCliente	
Descripción			
Tabla que almacena las direcciones de domicilio para un determinado Cliente.			
Nombre de Columna	Tipo de Dato	Descripción	
IdDirección	Int	Llave primaria auto incremental que identifica el registro de dirección dentro de la tabla.	
Dirección	Nvarchar(500)	Dirección exacta del domicilio del cliente.	
IdDistrito	Int	Campo de llave foránea que hace relación con la tabla TDistrito.	
IdCantón	Int	Campo de llave foránea que hace relación con la tabla TCantón.	
IdProvincia	Int	Campo de llave foránea que hace relación a la tabla TProvincia.	
Id_Cliente	Int	Campo de llave foránea que hace relación a la tabla de TClientes.	
Índices / Constraints			
Llave Primaria			
Nombre de Llave		Columnas	
PK_IdDireccion		IdDireccion	

Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TDireccionCliente_TDistrito_idDistrito	IdDistrito	TDistrito	IdDistrito	Relación foránea a la tabla TDistrito
FK_TDireccionCliente_TCanton_idCanton	IdCantón	TCantón	IdCantón	Relación foránea a la tabla de TCanton.
FK_TDireccionCliente_TProvincia_idProvincia	IdProvincia	TProvincia	IdProvincia	Relación foránea a la tabla de TProvincia.
FK_TDireccionCliente_TClientes_clientesIdCliente	IdCliente	TClientes	IdCliente	Relación foránea con la tabla TClientes.
Otros índices				
Nombre índice	Columnas		Descripción	

Fuente: Elaboración propia.

Tabla 43. Diccionario de datos, TTelefonoCliente.

Nombre de Tabla			TTelefonoCliente
Descripción			
Tabla que almacena los números de teléfono para un determinado cliente.			
Nombre de Columna	Tipo de Dato	Descripción	
IdTelefono	Int	Llave primaria auto incremental que identifica el registro de teléfono dentro de la tabla.	
Teléfono	Nvarchar(256)	Número de teléfono del cliente.	
Id_Cliente	Int	Campo de llave foránea que hace relación a la tabla de TClientes.	
Índices / Constraints			
Llave Primaria			
Nombre de Llave	Columnas		
PK_IdTelefono	IdTelefono		
Llaves foráneas			

Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TTelefonoCliente_TClientes_IdCliente	IdCliente	TClientes	IdCliente	Relación foránea con la tabla TClientes.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 44. Diccionario de datos, TCorreosClientes.

Nombre de Tabla		TCorreosClientes		
Descripción				
Tabla que almacena las direcciones de correo electrónico para un determinado cliente.				
Nombre de Columna	Tipo de Dato	Descripción		
IdCorreo	Int	Llave primaria auto incremental que identifica el registro de correo dentro de la tabla.		
Correo	Nvarchar(256)	Correo electrónico del cliente.		
Id_Cliente	Int	Campo de llave foránea que hace relación a la tabla de TClientes.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_IdCorreo		IdCorreo		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TTelefonoCliente_TClientes_IdCliente	IdCliente	TClientes	IdCliente	Relación foránea con la tabla TClientes.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 45. *Diccionario de datos,TCreditoClientes.*

Nombre de Tabla		TCreditoClientes
Descripción		
<p>Esta tabla se utiliza para almacenar la información de deudas en caso de que el cliente cuente con una deuda de crédito activa en esta tabla se pueden observar todos los detalles. Dicha tabla actúa como un resumen de la deuda del cliente.</p>		
Nombre de Columna	Tipo de Dato	Descripción
IdDeuda	Int	Almacena un numero auto incremental que identifica el registro de deuda. Esta corresponde a la llave primaria.
Deuda	Decimal	Almacena el total de deuda inicial que tiene el cliente.
Mensual	Decimal	Almacena el monto de pago mensual.
Cambio	Decimal	Almacena el cambio otorgado al cliente en el último pago realizado.
Ultimo_Pago	Decimal	Almacena el monto del último pago.
Fecha_Pago	DateTime	Almacena la fecha que corresponde al último pago de la deuda.
Deuda_Actual	Decimal	Almacena el monto de la deuda actual de la operación.
Fecha_Deuda	DateTime	Almacena la fecha de adquirida la deuda.
Ticket	Nvarchar(256)	Numero de tiquete generado para la deuda.
Fecha_Limite	DateTime	Almacena la fecha límite para realizar el siguiente pago a la deuda.
Id_Cliente	Int	Llave Foranea que liga el registro de deuda a un cliente existente de la tabla clientes.
Índices / Constraints		
Llave Primaria		
Nombre de Llave	Columnas	
PK_IdDeuda	IdDeuda	
Llaves foráneas		

Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Cedula_Cliente	Id_Cliente	TCientes	Id_Cliente	Relación con la tabla de clientes.
Otros índices				
Nombre índice	Columnas	Descripción		

Fuente: Elaboración propia.

Tabla 46. Diccionario de datos, THistoricoPagosClientes.

Nombre de Tabla		THistoricoPagosClientes
Descripción		
Esta tabla se utiliza para almacenar la información de pagos realizados de clientes a sus respectivas deudas activas dentro del sistema.		
Nombre de Columna	Tipo de Dato	Descripción
IdPagoCliente	Int	Almacena un numero auto incremental que identifica el registro de pago. Esta corresponde a la llave primaria.
Deuda	Decimal	Almacena el total de deuda inicial que tiene el cliente.
Pago	Decimal	Dinero recibido en la transacción de pago de deuda.
Mensual	Decimal	Almacena el monto de pago mensual.
Cambio	Decimal	Almacena el cambio otorgado al cliente en el pago.
Fecha_Pago	DateTime	Almacena la fecha que corresponde al pago de la deuda.
Deuda_Actual	Decimal	Almacena el monto de la deuda actual de la operación posterior al pago realizado.
Fecha_Deuda	DateTime	Almacena la fecha de adquirida la deuda.
Ticket	Nvarchar	Numero de tiquete generado para la deuda.

Fecha_Limite	DateTime	Almacena la fecha límite del próximo pago pendiente.		
DeudaAnterior	Decimal	Monto de la deuda, anterior al pago realizado.		
Id_Cliente	Int	Llave foránea que liga el registro de pago a un cliente existente de la tabla clientes.		
Id_Usuario	Int	Llave foránea que liga un registro de pago a un usuario de la tabla de usuarios.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_IdPagoCliente		IdPagoCliente		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Cliente	Id_Cliente	TClientes	Id_Cliente	Relación con la tabla TClientes.
FK_Id_Usuario	Id_Usuario	TUsers	ID	Relación con la tabla TUsers.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 47. Diccionario de datos, TProveedor.

Nombre de Tabla		TProveedor
Descripción		
Esta tabla se utiliza para almacenar la información básica de los proveedores que suministran los artículos al negocio.		
Nombre de Columna	Tipo de Dato	Descripción
Id_Proveedor	Int	Campo auto incremental que identifica el registro de proveedor en la tabla, este campo funciona como llave primaria.
Ced_Jur	BigInt	Almacena la cedula jurídica o física que identifica al proveedor.

Nombre_Proveedor	Nvarchar(256)	Almacena el nombre de la compañía o de la persona que identifica al proveedor.		
Fecha	DateTime	Fecha de registro al sistema del proveedor.		
Imagen	Varbinary(max)	Imagen asignada al registro de proveedor.		
Enabled	BIT	Este campo define si el proveedor fue borrado lógicamente de la base de datos.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_Proveedor		Id_Proveedor		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 48. *Diccionario de datos, TDireccionProveedor.*

Nombre de Tabla		TDireccionProveedor	
Descripción			
Tabla que almacena las direcciones de domicilio para un determinado Proveedor.			
Nombre de Columna		Tipo de Dato	Descripción
IdDirección		Int	Llave primaria auto incremental que identifica el registro de dirección dentro de la tabla.
Dirección		Nvarchar(500)	Dirección exacta del proveedor.
IdDistrito		Int	Campo de llave foránea que hace relación con la tabla TDistrito.
IdCantón		Int	Campo de llave foránea que hace relación con la tabla TCantón.

IdProvincia	Int	Campo de llave foránea que hace relación a la tabla TProvincia.		
Id_Proveedor	Int	Campo de llave foránea que hace relación a la tabla de TProveedor.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_IdDireccion		IdDireccion		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TDireccionCliente_TDistrito_idDistrito	IdDistrito	TDistrito	IdDistrito	Relación foránea a la tabla TDistrito
FK_TDireccionCliente_TCanton_idCanton	IdCantón	TCantón	IdCantón	Relación foránea a la tabla de TCanton.
FK_TDireccionCliente_TProvincia_idProvincia	IdProvincia	TProvincia	IdProvincia	Relación foránea a la tabla de TProvincia.
FK_TDireccionesProveedor_TProveedor_IdProveedor	Id_Proveedor	TProveedor	Id_Proveedor	Relación foránea con la tabla TProveedor.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 49. Diccionario de datos, TTelefonoProveedor.

Nombre de Tabla		TTelefonoProveedor
Descripción		
Tabla que almacena los números de teléfono para un determinado proveedor.		
Nombre de Columna	Tipo de Dato	Descripción
IdTelefono	Int	Llave primaria auto incremental que identifica el registro

		de teléfono dentro de la tabla.		
Teléfono	Nvarchar(256)	Número de teléfono del proveedor.		
Id_Proveedor	Int	Campo de llave foránea que hace relación a la tabla de TProveedor.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave	Columnas			
PK_IdTelefono	IdTelefono			
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TTelefonosProveedor_TProveedor_IdProveedor	Id_Proveedor	TProveedor	Id_Proveedor	Relación foránea con la tabla TProveedor.
Otros índices				
Nombre índice	Columnas		Descripción	

Fuente: Elaboración propia.

Tabla 50. Diccionario de datos, TCorreosProveedor.

Nombre de Tabla	TCorreosProveedor	
Descripción		
Tabla que almacena las direcciones de correo electrónico para un determinado proveedor.		
Nombre de Columna	Tipo de Dato	Descripción
IdCorreo	Int	Llave primaria auto incremental que identifica el registro de correo dentro de la tabla.
Correo	Nvarchar(256)	Correo electrónico del proveedor.
Id_Proveedor	Int	Campo de llave foránea que hace relación a la tabla de TProveedor.
Índices / Constraints		
Llave Primaria		
Nombre de Llave	Columnas	
PK_IdCorreo	IdCorreo	

Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TCorreosProveedor_TProveedor_IdProveedor	Id_Proveedor	TProveedor	Id_Proveedor	Relación foránea con la tabla TProveedor.
Otros índices				
Nombre índice	Columnas		Descripción	

Fuente: Elaboración propia.

Tabla 51. Diccionario de datos, TCreditoProveedor.

Nombre de Tabla			TCreditoProveedor
Descripción			
Esta tabla se utiliza para almacenar la información de deudas en caso de que el proveedor cuente con una deuda de crédito activa en esta tabla se pueden observar todos los detalles. Dicha tabla actúa como un resumen de la deuda del proveedor.			
Nombre de Columna	Tipo de Dato	Descripción	
IdCredito	Int	Almacena un numero auto incremental que identifica el registro de deuda. Esta corresponde a la llave primaria.	
Deuda	Decimal	Almacena el total de deuda inicial que tiene el cliente.	
Mensual	Decimal	Almacena el monto de pago mensual.	
Cambio	Decimal	Almacena el cambio otorgado al cliente en el último pago realizado.	
Ultimo_Pago	Decimal	Almacena el monto del último pago.	
Fecha_Pago	DateTime	Almacena la fecha que corresponde al último pago de la deuda.	
Deuda_Actual	Decimal	Almacena el monto de la deuda actual de la operación.	
Fecha_Deuda	DateTime	Almacena la fecha de adquirida la deuda.	

Ticket	Nvarchar(256)	Numero de tiquete generado para la deuda.		
Fecha_Limite	DateTime	Almacena la fecha límite para realizar el siguiente pago a la deuda.		
Id_Proveedor	Int	Llave Foranea que liga el registro de deuda a un proveedor existente de la tabla TProveedor.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_IdCredito		IdCredito		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TCreditoProveedor_TProveedor_IdProveedor	Id_Proveedor	TProveedor	Id_Proveedor	Relación con la tabla de TProveedor.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 52. Diccionario de datos, THistoricoPagosProveedor.

Nombre de Tabla		THistoricoPagosProveedor
Descripción		
Esta tabla se utiliza para almacenar la información de pagos realizados a proveedores en sus respectivas deudas activas dentro del sistema.		
Nombre de Columna	Tipo de Dato	Descripción
IdPago	Int	Almacena un numero auto incremental que identifica el registro de pago. Esta corresponde a la llave primaria.
Deuda	Decimal	Almacena el total de deuda inicial que tiene el cliente.
Pago	Decimal	Dinero recibido en la transacción de pago de deuda.
Mensual	Decimal	Almacena el monto de pago mensual.

Cambio	Decimal	Almacena el cambio otorgado al cliente en el pago.		
Fecha_Pago	DateTime	Almacena la fecha que corresponde al pago de la deuda.		
Deuda_Actual	Decimal	Almacena el monto de la deuda actual de la operación posterior al pago realizado.		
Fecha_Deuda	DateTime	Almacena la fecha de adquirida la deuda.		
Ticket	Nvarchar	Numero de tiquete generado para la deuda.		
Fecha_Limite	DateTime	Almacena la fecha límite del próximo pago pendiente.		
DeudaAnterior	Decimal	Monto de la deuda, anterior al pago realizado.		
Id_Proveedor	Int	Llave foránea que liga el registro de pago a un proveedor existente de la tabla TProveedor.		
Id_Usuario	Int	Llave foránea que liga un registro de pago a un usuario de la tabla de usuarios.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_IdPago		IdPago		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Proveedor	Id_Proveedor	TProveedor	Id_Proveedor	Relación con la tabla TProveedor.
FK_Id_Usuario	Id_Usuario	TUsers	ID	Relación con la tabla TUsers.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 53. *Diccionario de datos, TProducto.*

Nombre de Tabla	TProducto	
Descripción		
Esta tabla se utiliza para almacenar un producto en la base de datos.		
Nombre de Columna	Tipo de Dato	Descripción
Id_Producto	Int	Llave primaria, identifica el producto con un número único de registro.
Código_Producto	Int	Campo que identifica el código de producto con el que se consulta en una venta o compra.
Nombre_Producto	Nvarchar(30)	Almacena el Nombre con el que se identifica el producto.
Id_Tipo_Producto	Int	Llave foránea a la tabla tipo Producto.
Cantidad_Producto	Int	Almacena la cantidad del producto en existencias.
Precio_Venta	Decimal	Almacena el precio unitario del producto con el que se ofrece al cliente. Precio no contiene impuestos ni descuentos.
Precio_Costo	Decimal	Almacena el precio en el cual se adquiere al cliente. Precio no contiene impuestos ni descuentos.
Descuento de Producto	Decimal	Almacena el porcentaje de descuento del producto.
Descripción_Producto	Nvarchar(300)	Almacena la Descripción del producto.
Fecha	DateTime	Fecha de Ingreso de Producto al sistema.
Imagen	VarBinary	Imagen asignada al producto.

Enabled	BIT	Este campo define si el proveedor fue borrado lógicamente de la base de datos.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_Producto		Id_Producto		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Tipo_Producto	Id_Tipo_Producto	TTipoProducto	Id_TipoProducto	Hace la relación con la tabla de Tipo Producto.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 54. Diccionario de datos, TTipoProducto.

Nombre de Tabla		TTipoProducto		
Descripción				
Esta tabla se utiliza para almacenar los distintos tipos de producto que vende el negocio.				
Nombre de Columna	Tipo de Dato		Descripción	
Id_TipoProducto	Int		Llave primaria, auto incremental que identifica el tipo de producto dentro de la tabla.	
Nombre_Tipo_Producto	Nvarchar		Almacena el nombre que identifica el tipo de producto.	
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_TipoProducto		Id_TipoProducto		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
Otros índices				

Nombre índice	Columnas	Descripción

Fuente: Elaboración propia.

Tabla 55. Diccionario de datos, TRegistroCompras.

Nombre de Tabla		TRegistroCompras
Descripción		
Esta tabla almacena el registro de compras en el sistema		
Nombre de Columna	Tipo de Dato	Descripción
Id_RegistroCompras	Int	Llave primaria, auto incremental que identifica el registro de compra dentro de la tabla.
Numero_Factura_Proveedor	Nvarchar(50)	Almacena el numero de la factura otorgado por el proveedor.
Id_Proveedor	Int	Llave foránea que hace relación con el proveedor al que se realiza la compra.
Total_bruto	Decimal	Monto total bruto de la compra.
Total_descuento	Decimal	Descuento total aplicado en la compra.
Total_IVA	Decimal	Impuesto total aplicando en la compra.
Total_netto	Decimal	Monto total Neto de la compra.
Fecha_Compra	DateTime	Fecha en la que se realiza la compra al proveedor.
Cambio_Compra	Decimal	Almacena el cambio devuelto en la operación de compra.
Dinero_Recibido	Decimal	Almacena el dinero que recibió el proveedor en el pago de la compra.
Método_Pago	Int	Almacena un numero identificador al método de pago donde 1 es crédito, 2 efectivo, 3 tarjeta y 4 transferencia bancaria.

ReferenciaBancaria	Nvarchar(45)	Almacena el número de referencia bancaria en caso de transacción por tarjeta o transferencia.		
Id_User	Int	Campo de llave foránea que relaciona con la tabla TUser.		
Estado_Compra	Boolean	Campo que identifica si un registro de compra ya fue tomando en cuenta en un proceso de cierre.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_RegistroCompras		Id_RegistroCompras		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_TRegistroCompras_TProveedor_Id_Proveedor	Id_Proveedor	TProveedor	Id_Proveedor	Hace la relación con la tabla TProveedor.
FK_TUser	Id_User	TUser	Id	Hace la relación con la tabla TUser.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 56. Diccionario de datos, TDetallesCompras.

Nombre de Tabla	TDetallesCompras	
Descripción		
Esta tabla almacena los detalles de productos adquiridos en cada compra.		
Nombre de Columna	Tipo de Dato	Descripción
Id_detalleCompra	Int	Llave Primaria auto incremental, identifica el registro detalle compra dentro de la tabla.
Id_Producto	Int	Llave foránea a la tabla producto donde

		especifica el producto que se está comprando.		
Id_RegistroCompra	Int	Llave foránea a la tabla Registro Compra donde identifica dicho detalle a una compra.		
Cantidad_Producto	Int	Almacena la cantidad de artículos adquiridos en esa compra.		
Monto_Bruto_Detalle	Decimal	Almacena el monto bruto del detalle de ese producto.		
Descuento_Detalle	Decimal	Almacena el descuento aplicado para ese producto.		
Impuesto_Detalle	Decimal	Almacena el impuesto aplicado para ese producto.		
Monto_Neto_Detalle	Decimal	Almacena el monto neto del detalle del producto.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave	Columnas			
PK_Id_detalleCompra	Id_detalleCompra			
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Producto	Id_Producto	TProducto	Id_Producto	Hace la relación con la tabla de producto.
FK_Id_RegistroCompra	Id_RegistroCompra	TRegistroDeCompras	Id_RegistroDeCompras	Hace la relación con la tabla de compras.
Otros índices				
Nombre índice	Columnas		Descripción	

Fuente: Elaboración propia.

Tabla 57. Diccionario de datos, TTemporalCompras.

Nombre de Tabla		TTemporalCompras		
Descripción				
<p>Esta tabla almacena los detalles de producto en una compra de forma temporal previo de procesar la compra, esta tabla tiene la funcionalidad de preparar todos los productos previamente antes de procesarlos a la tabla Detalles compras.</p>				
Nombre de Columna	Tipo de Dato	Descripción		
Id_TempCompras	Int	Llave Primaria auto incremental, identifica el registro de temporal compras dentro de la tabla.		
Id_Producto	Int	Llave foránea a la tabla producto donde especifica el producto que se está comprando.		
Id_Proveedor	Int	Llave foránea a la tabla TProveedor.		
Id_User	Int	Llave foránea a la tabla TUser.		
Cantidad_Producto	Int	Almacena la cantidad de artículos adquiridos en esa compra.		
Monto_Bruto_Detalle	Decimal	Almacena el monto bruto del detalle de ese producto.		
Descuento_Detalle	Decimal	Almacena el descuento aplicado para ese producto.		
Impuesto_Detalle	Decimal	Almacena el impuesto aplicado para ese producto.		
Fecha	DateTime	Almacena la fecha del registro temporal de compra.		
Monto_Neto_Detalle	Decimal	Almacena el monto neto del detalle del producto.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave	Columnas			
PK_Id_TempCompras	Id_TempCompras			
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Producto	Id_Producto	TProducto	Id_Producto	Hace la relación con la tabla de producto.

FK_Id_Proveedor	Id_Proveedor	TProveedor	Id_Proveedor	Hace la relación con la tabla de TProveedor.
FK_Id_User	Id_User	TUser	Id_User	Hace la relación con la tabla TUser.
Otros índices				
Nombre índice		Columnas		Descripción

Fuente: Elaboración propia.

Tabla 58. Diccionario de datos, TRegistroDeVentas.

Nombre de Tabla			TRegistroDeVentas
			Descripción
Esta tabla almacena la información realizada a los registros de ventas en el sistema.			
Nombre de Columna	Tipo de Dato	Descripción	
Id_RegistroVenta	Int	Llave Primaria, auto incremental, identifica la venta dentro de la tabla.	
Id_Usuario	Int	Llave foránea a la tabla de usuario, de esta forma identificando el dependiente que realiza la venta.	
Id_Cliente	Int	Llave foránea a la tabla de clientes, identificando a cuál cliente se le está realizando la venta.	
Fecha_Venta	DateTime	Almacena la fecha cuando se realiza la venta.	
Total_bruto	Decimal	Monto total bruto de la compra.	
Total_descuento	Decimal	Descuento total aplicado en la compra.	
Total_IVA	Decimal	Impuesto total aplicando en la compra.	
Total_netto	Decimal	Monto total Neto de la compra.	
Cambio_Venta	Decimal	Almacena el cambio devuelto en la operación de compra.	
Dinero_Recibido	Decimal	Almacena el dinero que recibió el negocio en el pago de la venta.	

Método_Pago	Int	Almacena un numero identificador al método de pago donde 1 es crédito, 2 efectivo, 3 tarjeta y 4 transferencia bancaria.		
ReferenciaBancaria	Nvarchar(45)	Almacena el número de referencia bancaria en caso de transacción por tarjeta o transferencia.		
Estado_Venta	Boolean	Campo que identifica si un registro de venta ya fue tomando en cuenta en un proceso de cierre.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_RegistroVenta		Id_RegistroVenta		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Usuario	Id_Usuario	TUsuarios	ID	Hace la relación con la tabla de usuarios.
FK_Id_Cliente	Id_Cliente	TClientes	Id_Cliente	Hace la relación con la tabla de clientes.
Otros índices				
Nombre índice		Columnas	Descripción	

Fuente: Elaboración propia.

Tabla 59. Diccionario de datos, TDetallesVentas.

Nombre de Tabla		TDetallesVentas
Descripción		
Esta tabla almacena los detalles de línea para cada venta.		
Nombre de Columna	Tipo de Dato	Descripción
Id_detalleVenta	Int	Llave Primaria auto incremental, identifica el registro detalle venta dentro de la tabla.
Id_Producto	Int	Llave foránea a la tabla producto donde

		especifica el producto que se está vendiendo.		
Id_RegistroVenta	Int	Llave foránea a la tabla Registro Venta donde identifica dicho detalle a una venta.		
Cantidad_Venta	Int	Almacena la cantidad de artículos vendidos.		
Monto_Bruto_Detalle	Decimal	Almacena el monto bruto del detalle de ese producto.		
Descuento_Detalle	Decimal	Almacena el descuento aplicado para ese producto.		
Impuesto_Detalle	Decimal	Almacena el impuesto aplicado para ese producto.		
Monto_Neto_Detalle	Decimal	Almacena el monto neto del detalle del producto.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_detalleVenta		Id_detalle_Venta		
Claves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Producto	Id_Producto	TProducto	IdProducto	Hace la relación con la tabla de Producto.
FK_Id_RegistroVenta	Id_RegistroVenta	TRegistroDeVentas	Id_RegistroVenta	Hace la relación con la tabla Registro Ventas
Otros índices				
Nombre índice		Columnas		Descripción

Fuente: Elaboración propia.

Tabla 60. Diccionario de datos, TTemporalVentas.

Nombre de Tabla		TTemporalVentas		
Descripción				
Esta tabla almacena los detalles de producto en una venta de forma temporal previo de procesar la venta, esta tabla tiene la funcionalidad de preparar todos los productos previamente antes de procesarlos a la tabla Detalles ventas.				
Nombre de Columna	Tipo de Dato	Descripción		
Id_TempVentas	Int	Llave Primaria auto incremental, identifica el registro de temporal compras dentro de la tabla.		
Id_Producto	Int	Llave foránea a la tabla producto donde especifica el producto que se está comprando.		
Id_Cliente	Int	Llave foránea a la tabla TCliente.		
Id_User	Int	Llave foránea a la tabla TUser.		
Cantidad_Producto	Int	Almacena la cantidad de artículos adquiridos en esa compra.		
Monto_Bruto_Detalle	Decimal	Almacena el monto bruto del detalle de ese producto.		
Descuento_Detalle	Decimal	Almacena el descuento aplicado para ese producto.		
Impuesto_Detalle	Decimal	Almacena el impuesto aplicado para ese producto.		
Fecha	DateTime	Almacena la fecha del registro temporal de venta.		
Monto_Neto_Detalle	Decimal	Almacena el monto neto del detalle del producto.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_detalleVenta		Id_TempVentas		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Producto	Id_Producto	TProducto	Id_Producto	Hace la relación con la tabla de producto.
FK_Id_Cliente	Id_Cliente	TCliente	Id_Cliente	Hace la relación con la

				tabla de TCliente.
FK_Id_User	Id_User	TUser	Id_User	Hace la relación con la tabla TUser.
Otros índices				
Nombre índice	Columnas	Descripción		

Fuente: Elaboración propia.

Tabla 61. Diccionario de datos, TRegistroCierre

Nombre de Tabla		TRegistroCierre
Descripción		
Esta tabla contiene la información histórica de los cierres diarios de Caja		
Nombre de Columna	Tipo de Dato	Descripción
Id_Cierre	Int	Llave Primaria auto incremental identifica el registro de cierre dentro de la tabla.
Id_Usuario	Int	Llave foránea identifica el usuario que realiza el cierre de caja.
Fecha_Cierre	DateTime	Fecha en la que se realiza el cierre de caja.
Dinero_En_Cajas	Decimal	Cantidad de Dinero en efectivo en caja.
Dinero_En_Cuentas	Decimal	Cantidad de Dinero en cuentas bancarias.
Total_Ventas_Efectivo	Decimal	Monto de ventas en efectivo realizadas durante el día de operación.
Total_Ventas_Cuentas	Decimal	Monto de ventas realizadas vía transferencia bancaria durante el día de operación.
Total_Ventas_Crédito	Decimal	Monto de ventas realizadas a crédito durante el día de operación.
Total_Compras_Efectivo	Decimal	Monto de compras en efectivo realizadas durante el día de operación.

Total_Compras_Cuentas	Decimal	Monto de ventas realizadas vía transferencia bancaria durante el día de operación.		
Total_Compras_Credito	Decimal	Monto de ventas realizadas a crédito durante el día de operación.		
Monto_Faltante_Cuentas	Decimal	Monto que indica si existe algún faltante de dinero en el arqueo de cuentas.		
Monto_Faltante_Cajas	Decimal	Monto que indica si existe algún faltante de dinero en el arqueo de caja.		
Monto_Sobrante_Cuentas	Decimal	Monto que indica si existe algún sobrante de dinero en el arqueo de cuentas.		
Monto_Sobrante_Cajas	Decimal	Monto que indica si existe algún sobrante de dinero el arqueo de cajas.		
Id_Apertura	Int	Llave foránea que realiza la relación de un registro de cierre con su respectiva apertura.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_Cierre		Id_Cierre		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Usuario	Id_Usuario	TUsuarios	ID	Hace la relación con la tabla de usuarios.
FK_Id_Apertura	Id_Apertura	TRegistroApertura	Id_Apertura	Relación con la tabla de apertura.
Otros índices				
Nombre índice	Columnas		Descripción	

Fuente: Elaboración propia.

Tabla 62. Diccionario de datos, TRegistroApertura.

Nombre de Tabla		TRegistroApertura		
Descripción				
Esta tabla contiene la información histórica de las aperturas de caja.				
Nombre de Columna	Tipo de Dato	Descripción		
Id_Apertura	Int	Llave Primaria auto incremental identifica el registro de apertura dentro de la tabla.		
Id_Usuario	Int	Llave foránea identifica el usuario que realiza el cierre de caja.		
Fecha_Apertura	DateTime	Fecha en la que se realiza la apertura de caja.		
Dinero_En_Cajas	Decimal	Cantidad de Dinero en efectivo en caja.		
Dinero_En_Cuentas	Decimal	Cantidad de Dinero en cuentas bancarias.		
Índices / Constraints				
Llave Primaria				
Nombre de Llave		Columnas		
PK_Id_Cierre		Id_Cierre		
Llaves foráneas				
Nombre de Llave	Columna	Tabla foránea	Columna foránea	Descripción
FK_Id_Usuario	Id_Usuario	TUsuarios	ID	Hace la relación con la tabla de usuarios.
Otros índices				
Nombre índice	Columnas	Descripción		

Fuente: Elaboración propia.

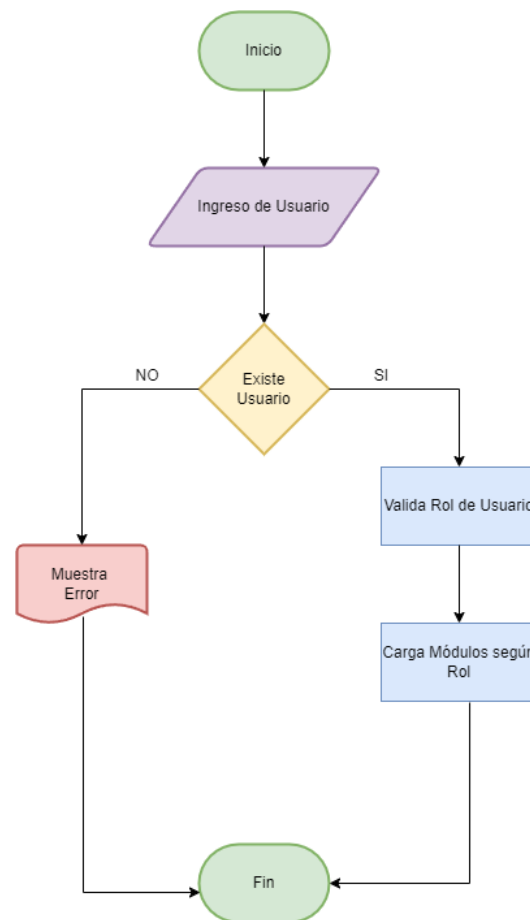
Diagramas de Procesos

Los diagramas de procesos muestran los algoritmos que toman las principales funcionalidades del prototipo, es una forma gráfica para poder entender de mejor manera el comportamiento de un sistema. Los principales procesos identificados en el Prototipo funcional para el manejo de ventas para Librería y Bazar Dieka ubicado en Moravia, son los procesos de ingreso, modificación, eliminación de los registros de las distintas entidades

del prototipo, los procesos de venta y compra, y los procesos de operaciones de caja. A continuación, se muestra de forma gráfica cada uno de estos procesos.

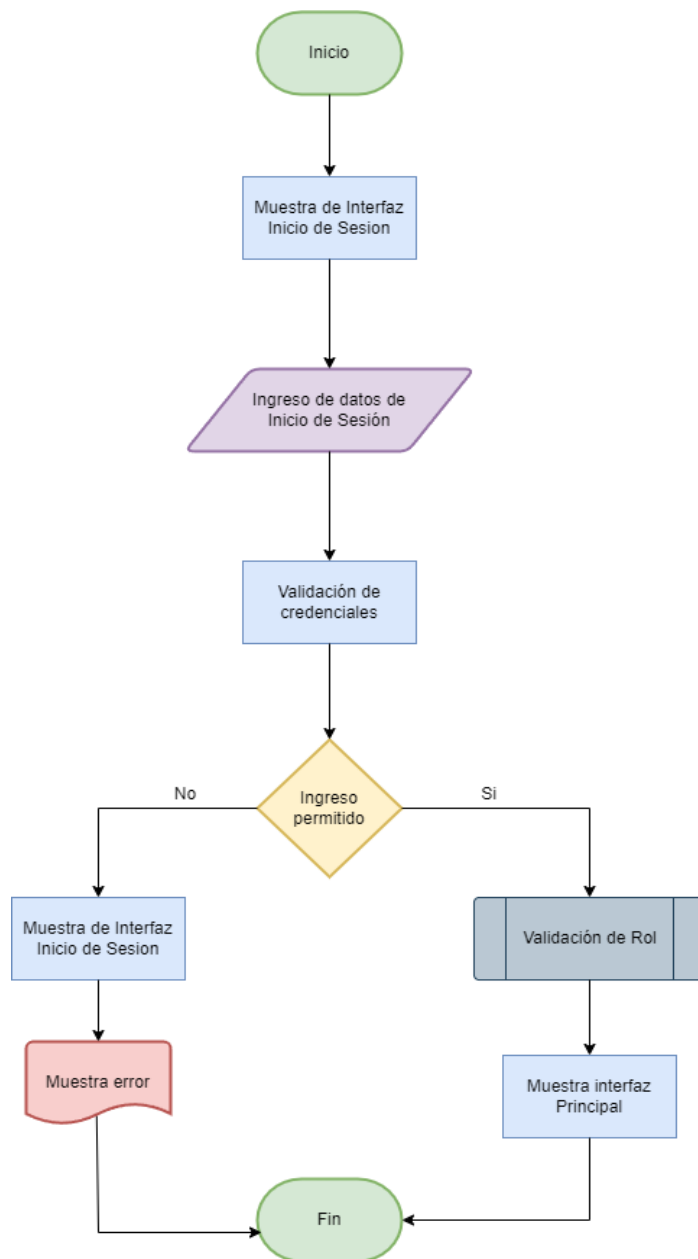
El siguiente proceso es utilizado en todos los módulos y tiene la funcionalidad de validar el rol del usuario.

Figura 32. Diagrama de validación de rol



Fuente: Elaboración Propia.

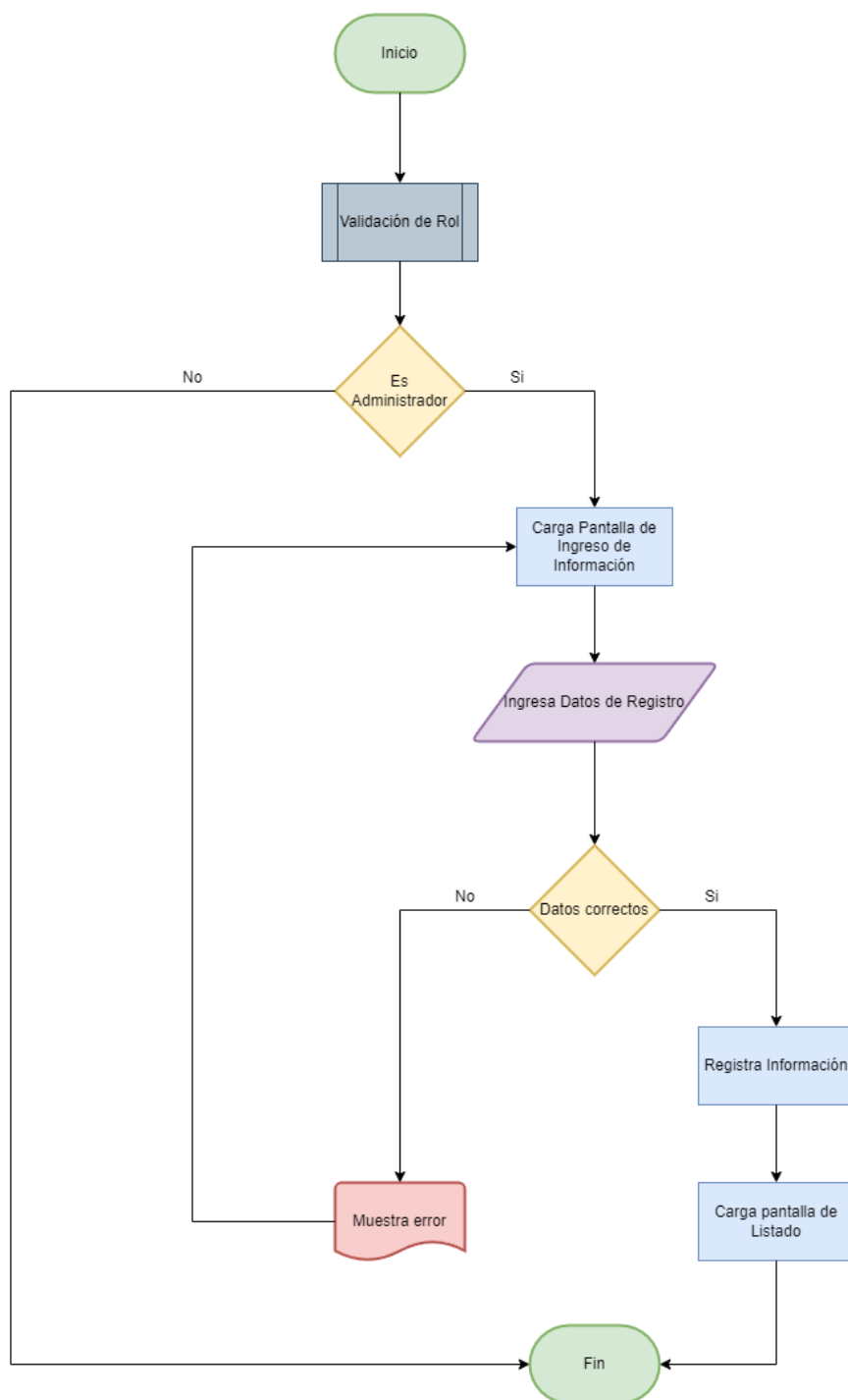
El siguiente proceso se encarga del inicio de sesión dentro del sistema. Es requerido que el usuario cuente de un usuario y una contraseña para hacer uso de todas las funcionalidades del prototipo.

Figura 33. Diagrama de proceso de inicio de sesión

Fuente: Elaboración propia.

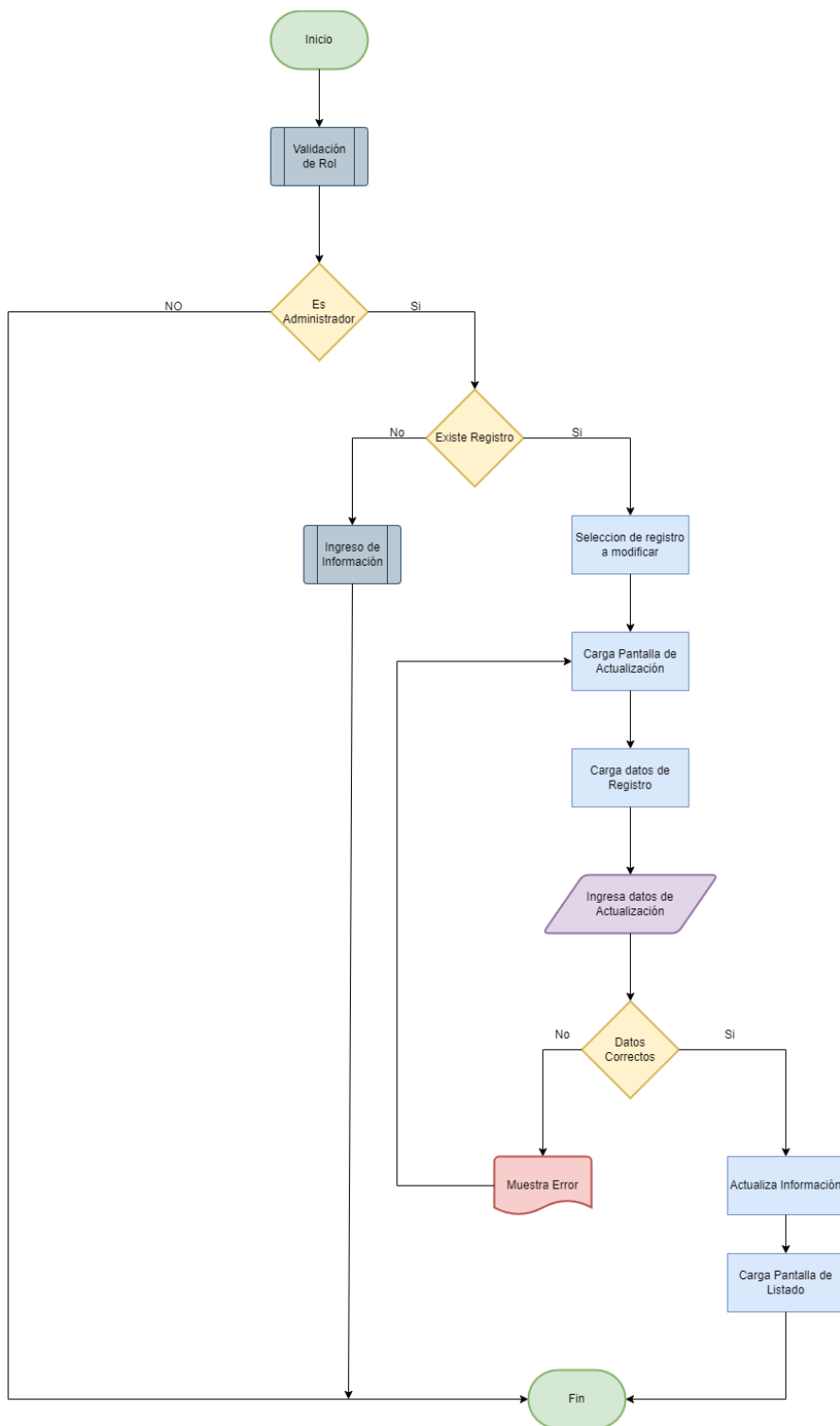
Los siguientes procesos se encargan de la manipulación de los datos en los distintos módulos del prototipo, dichos procesos encapsulan las operaciones de ingreso, actualización, y borrado de registros.

Figura 34. Diagrama de proceso ingreso de información.



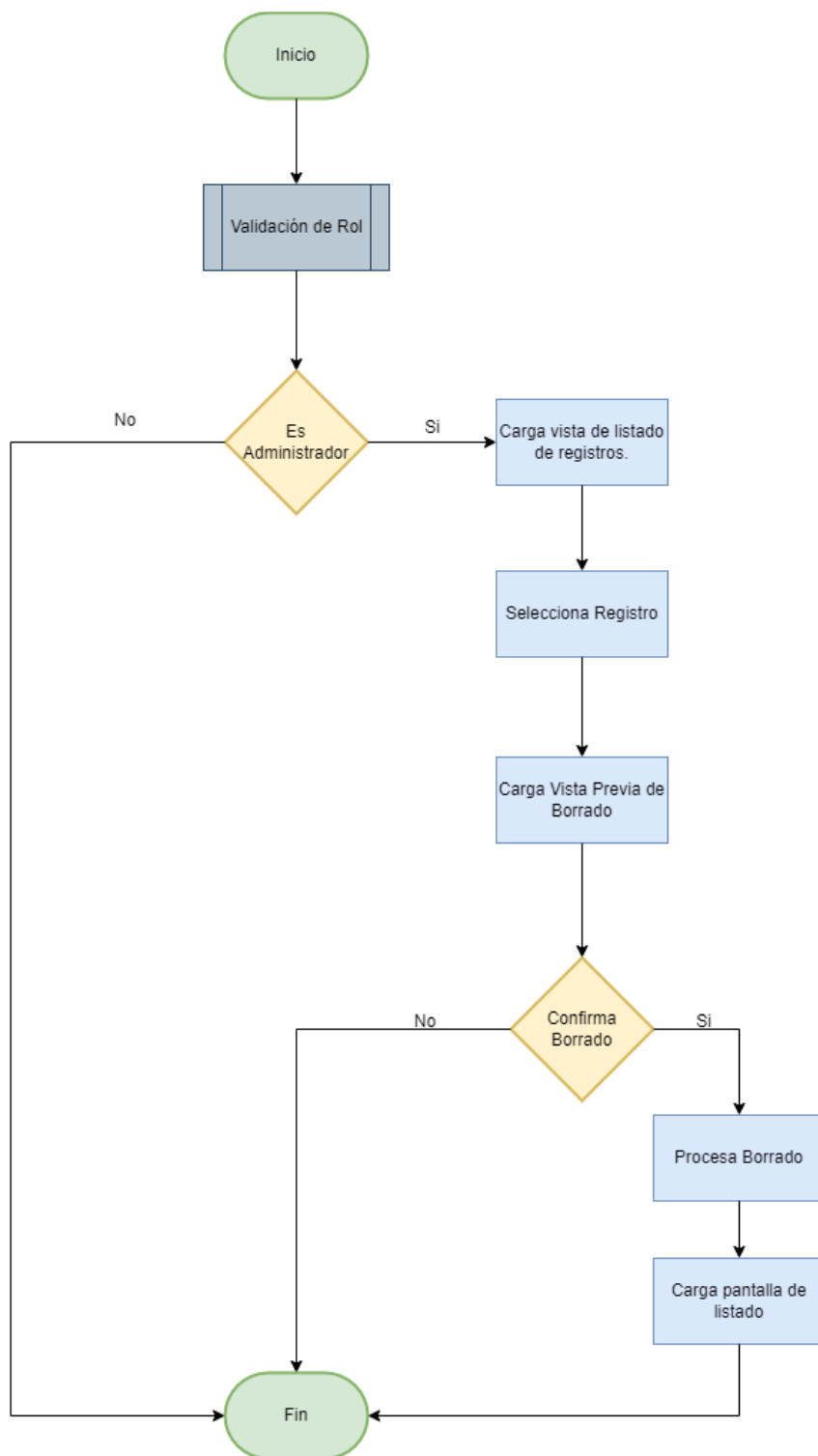
Fuente: Elaboración propia.

Figura 35. Diagrama de proceso de actualización de información.



Fuente: Elaboración propia.

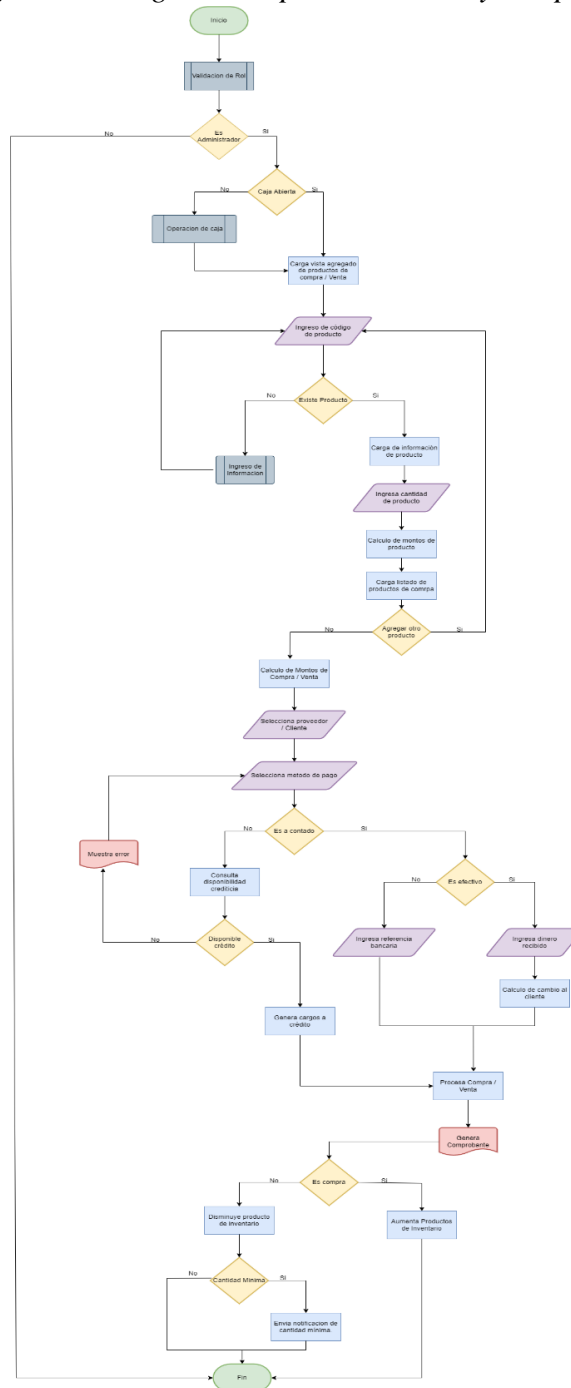
Figura 36. Diagrama de Proceso de borrado de información.



Fuente: Elaboración propia.

El siguiente es el proceso más importante del prototipo ya que encapsula la naturaleza del sistema, estos son el proceso de ventas y compras. Dicho proceso se encarga de realizar el ingreso y salida tanto de dinero como de productos al sistema.

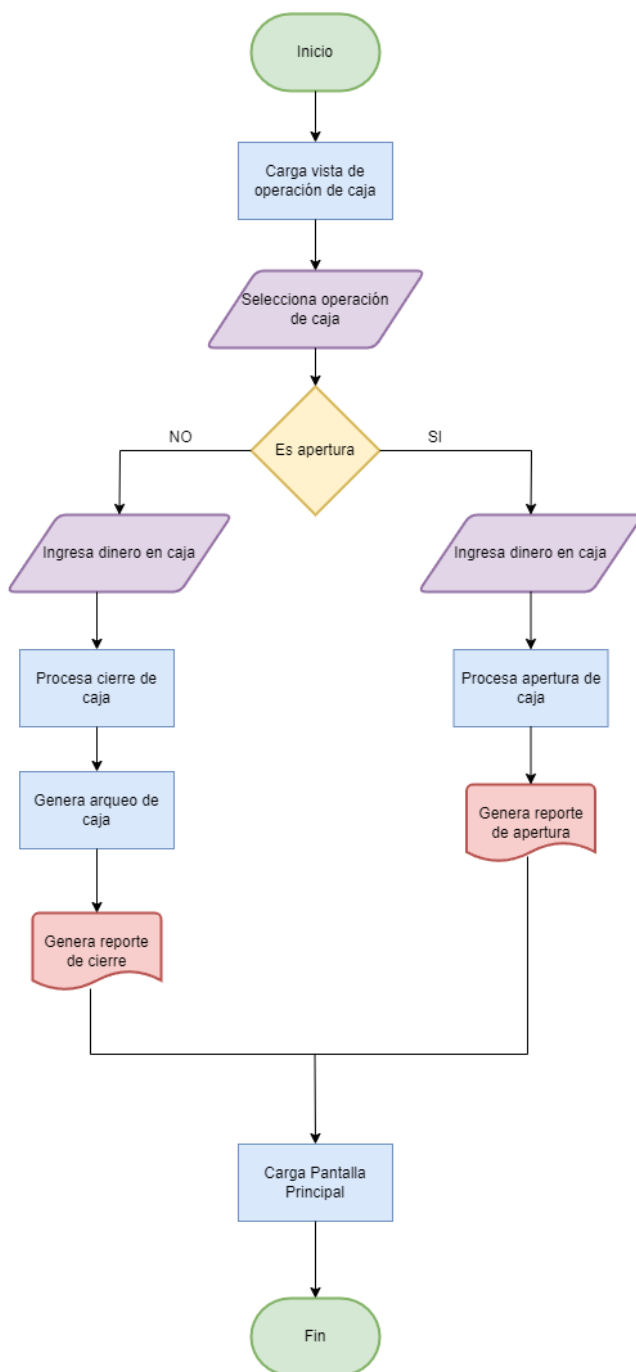
Figura 37. Diagrama de proceso ventas y compras.



Fuente: Elaboración propia.

Por último, se menciona el proceso de operaciones de caja el cual se encarga de abrir una caja para dar comienzo a las operaciones de compra y venta, y el cierre de caja para totalizar los montos de dinero al final del día de operaciones.

Figura 38. Diagrama de Proceso cierre de caja.



Fuente: Elaboración propia.

Diseño de salidas

Las salidas son aquellos resultados que se le muestra al usuario al finalizar una operación dentro del prototipo. El diseño de salidas del presente prototipo se realiza en pantalla, en reporte estilo PDF, y en formato de impresión para el caso de los “tickets”.

Diseño de salidas en pantalla

La siguiente salida en pantalla se utiliza tanto en los módulos de compras y ventas a créditos y se muestra el detalle de un pago realizado a la operación de crédito.

Figura 39. *Diseño de pantalla detalle de pago.*

Sandra Mendoza Rosales

Reporte de Pago

Deuda ¢1 000 000,00

Fecha deuda: 01/ene./001

Pago: ¢60 000,00

Fecha del pago: 03/ene./2023

Monto cuota por mes: ¢50 000,00

Deuda Actual: ¢800 000,00

Ticket: 0000000004

Usuario: Diegos Mendez Mendez

[Ticket](#)

Fuente: Elaboración propia.

La siguiente salida se utiliza en los módulos de compras y ventas que muestra el detalle de la compra o de la venta y los productos involucrados en la operación.

Figura 40. *Diseño de pantalla reporte de compra.*

Reporte de Compra						
Fecha:	29/1/2023 08:48:11	Id de Compra:	9			
Nombre de Proveedor:	Studmark	Numero de Factura Proveedor:	45680			
Metodo de Pago:	0					
Listado de Productos						
Código de Producto	Nombre de Producto	Precio de Costo	Cantidad	Total Impuesto	Total Descuento	Total Neto
1	Borrador	€50,00	20	€130,00	€20,00	€1 000,00
1	Borrador	€50,00	40	€260,00	€40,00	€2 000,00
1	Borrador	€50,00	50	€325,00	€50,00	€2 500,00
2	Lapiz	€100,00	50	€650,00	€0,00	€5 000,00
1	Borrador	€50,00	50	€325,00	€50,00	€2 500,00
Resultados del 1 al 2 de 7 /5						
0 1 2 Siguiente » Ultimo »						
					Total Descuentos:	€240,00
					Total Impuestos:	€2 600,00
					Total Neto:	€22 840,00

Fuente: Elaboración propia.

Diseño de salidas en formato PDF

La siguiente salida es un reporte en formato PDF que muestra el listado de clientes registrados en el sistema:

Figura 41. *Diseño de Salida de Reporte*



Cedula	Name	Phone	Email
999999999	Raúl	88888888	correo9000@gmail.com
888888880	Sandra	88888888	correo4@gmail.com
789547856	Pedro	88888888	correo9000@gmail.com
123942829	Rosaura	88888888	correo9001@gmail.com
123412341	Mauro	88888888	correo91@gmail.com
987623123	Arturo	88888888	correo91@gmail.com
1234123421	Maria	78547854	correo91@gmail.com
987231234123	Juan	71232123	correo9001@gmail.com
521361896212	Ramon	99999992	correocorreo@correo.com
961273821	Saul	71623812	correo223@gmail.com

Fuente: Elaboración propia.

En la siguiente imagen se puede observar el reporte de pagos realizados de un determinado cliente:

Figura 42. *Diseño de salida de pago de cliente.*



Reporte de Pagos para cedula de cliente:
888888880

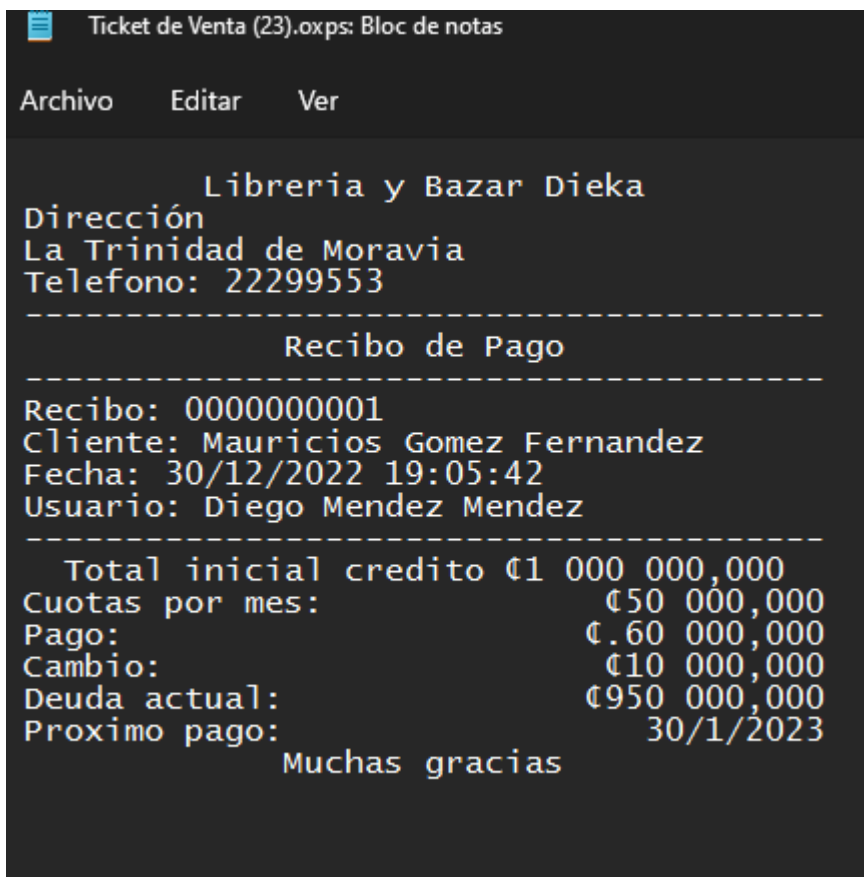
Deuda Inicial	DeudaActual	Pago Recibido	Fecha
1000000,00	800000,00	60000,00	3/1/2023 20:10:55
1000000,00	850000,00	100000,00	28/12/2022 08:35:57
1000000,00	900000,00	50000,00	28/12/2022 08:35:01
1000000,00	950000,00	60000,00	28/12/2022 08:34:45

Fuente: Elaboración propia.

Diseño de salidas en formato de Impresión

En la siguiente imagen se muestra el ticket generado y que se le entrega al cliente cuando realiza un pago a un crédito. Este se envía a la impresora que tenga configurado el sistema.

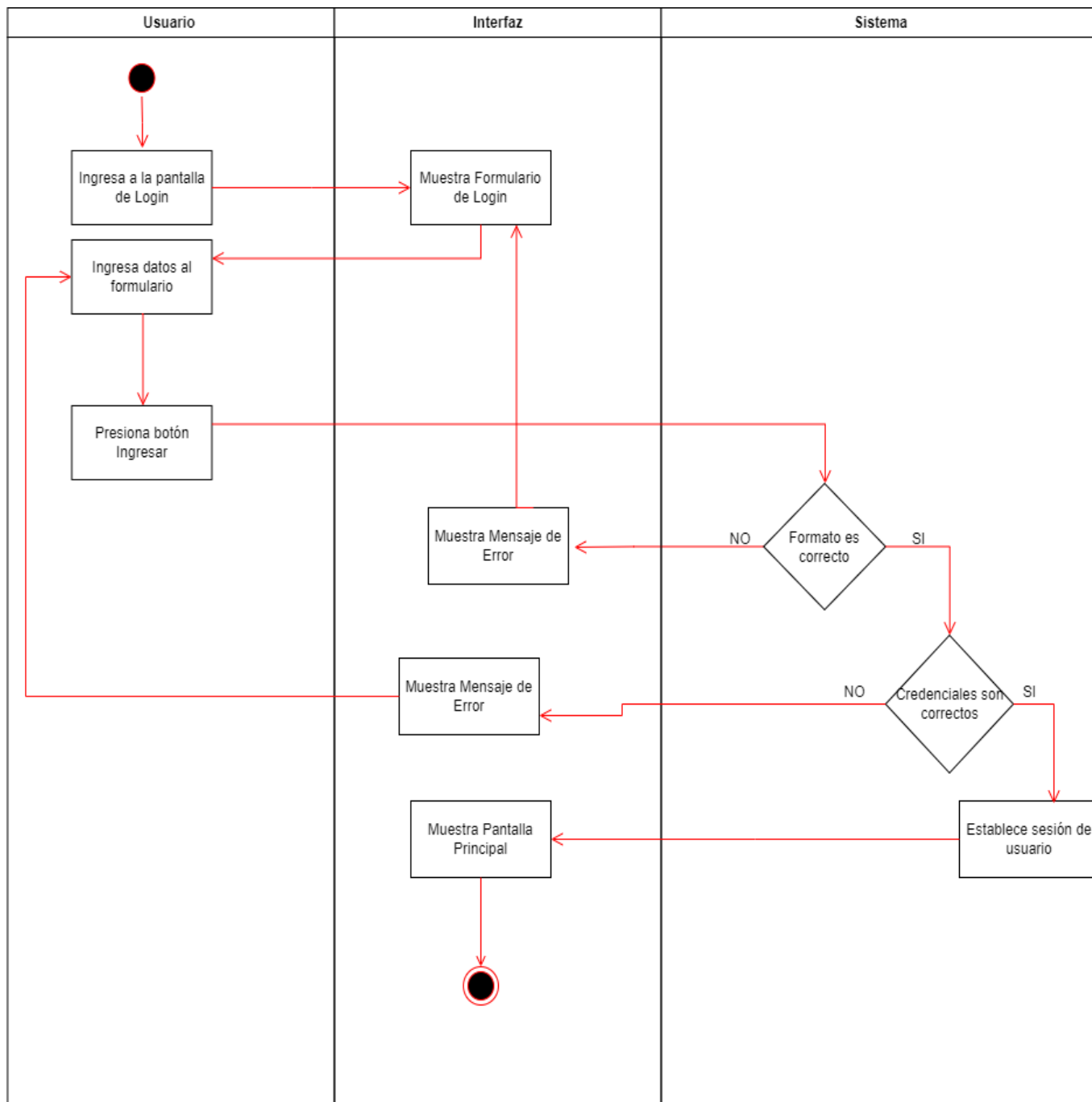
Figura 43. *Formato de ticket.*



Fuente: Elaboración propia.

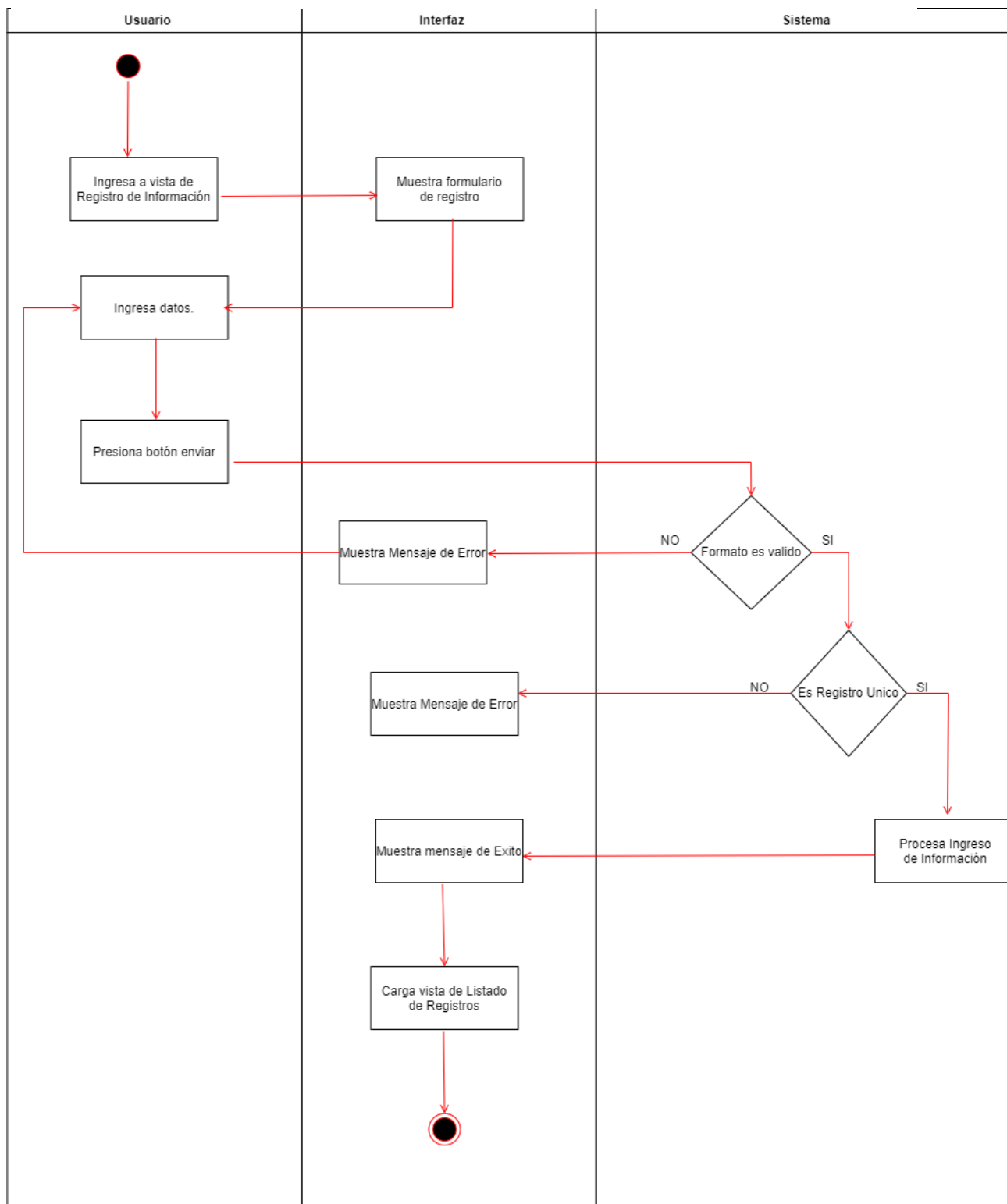
Diagramas de Actividades

Figura 45. Diagrama de Actividades inicio de sesión.



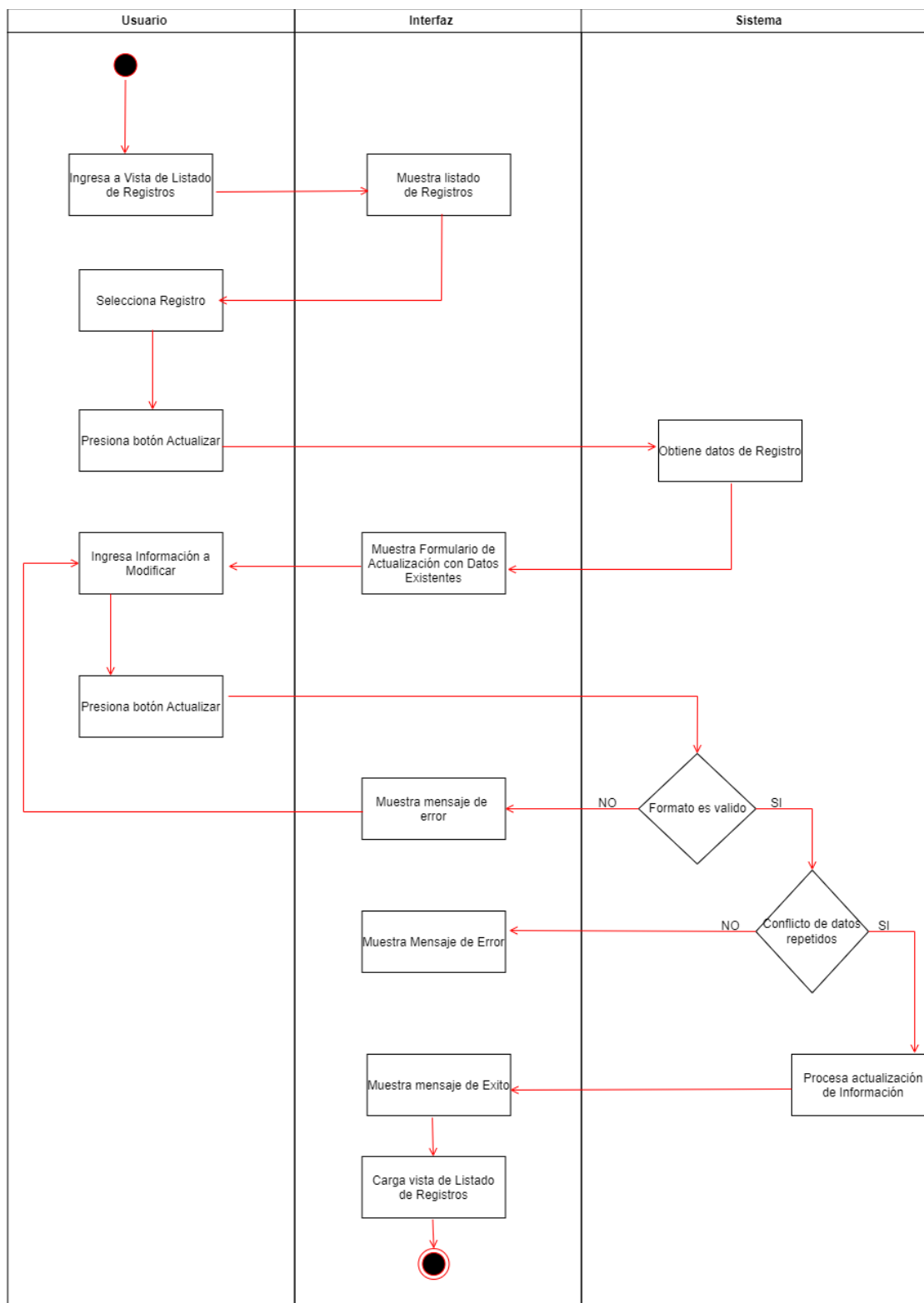
Fuente: Elaboración Propia.

Figura 46.Diagrama de actividades registro de información.



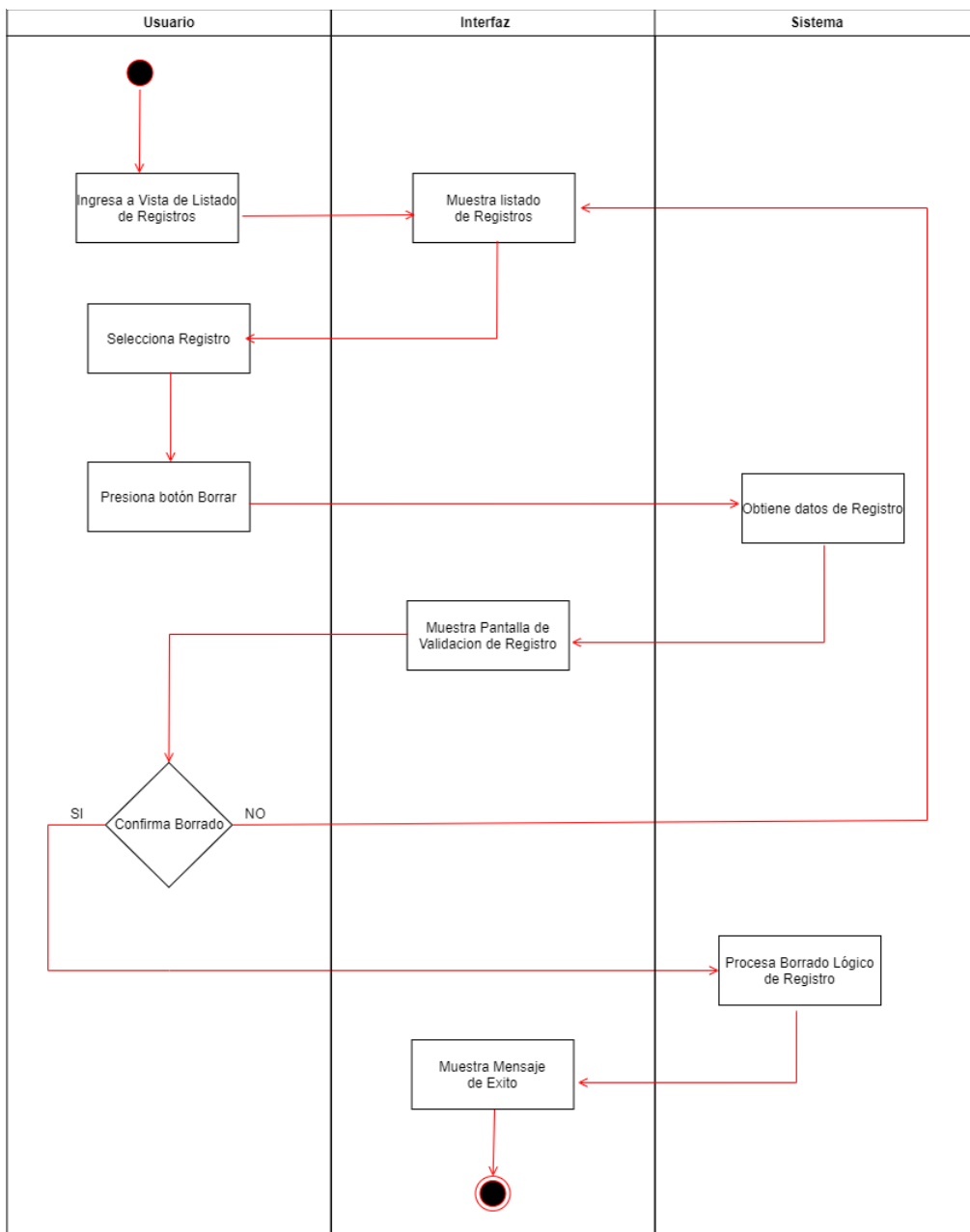
Fuente: Elaboración propia.

Figura 47. Diagrama de Actividades actualizado de información.



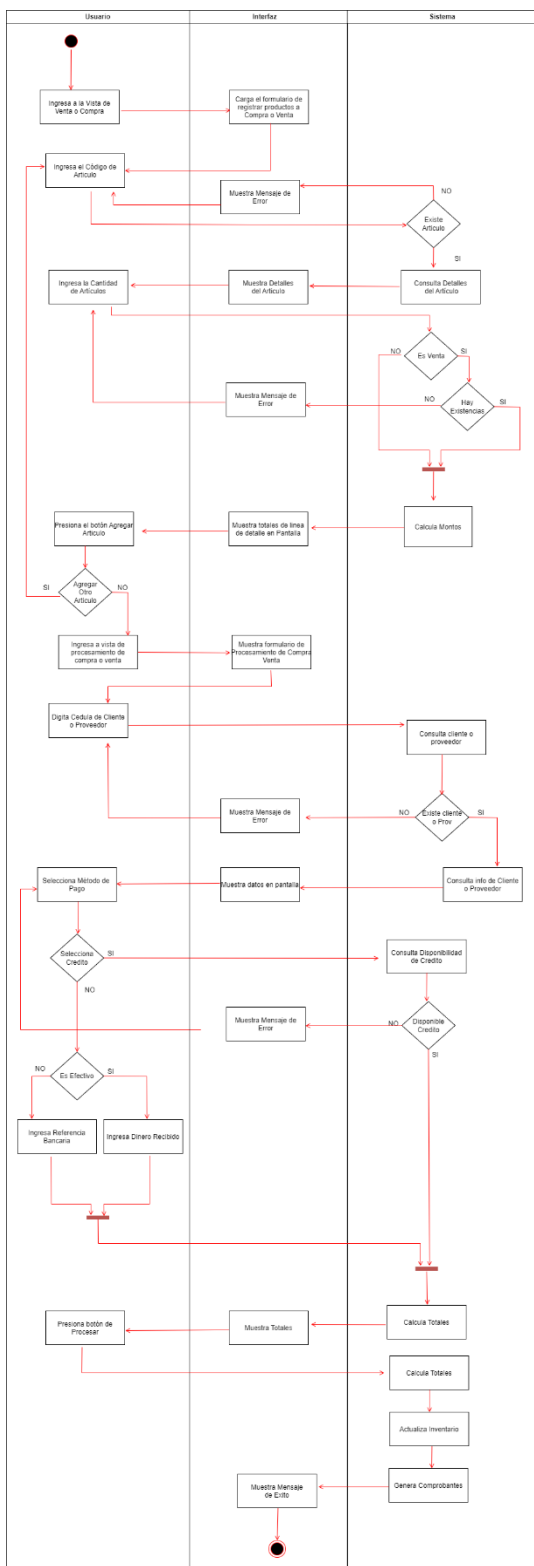
Fuente: Elaboración propia.

Figura 48. Diagrama de actividades de borrado de información.



Fuente: Elaboración propia.

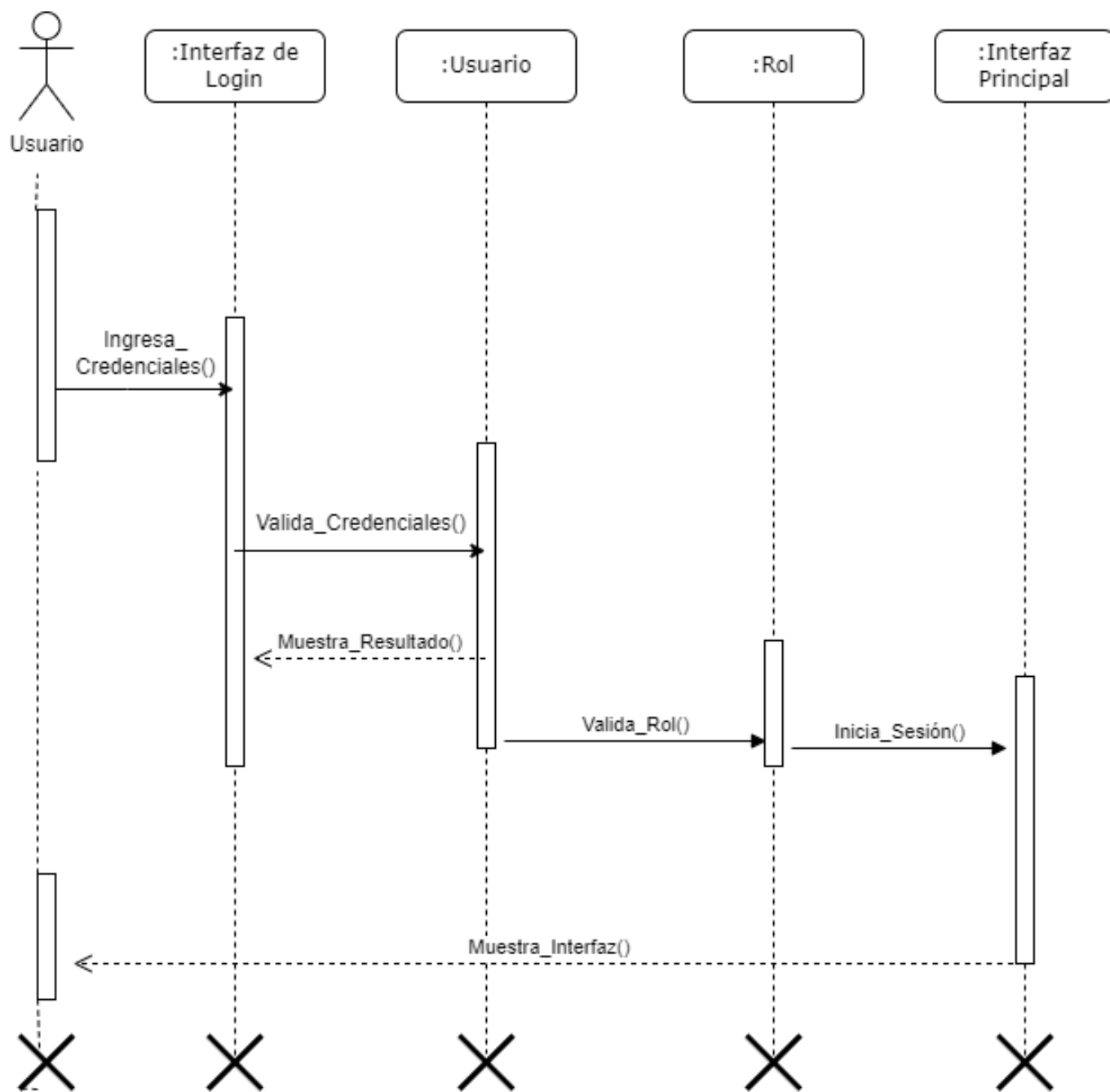
Figura 49. Diagrama de actividades de compra y venta.



Fuente: Elaboración propia.

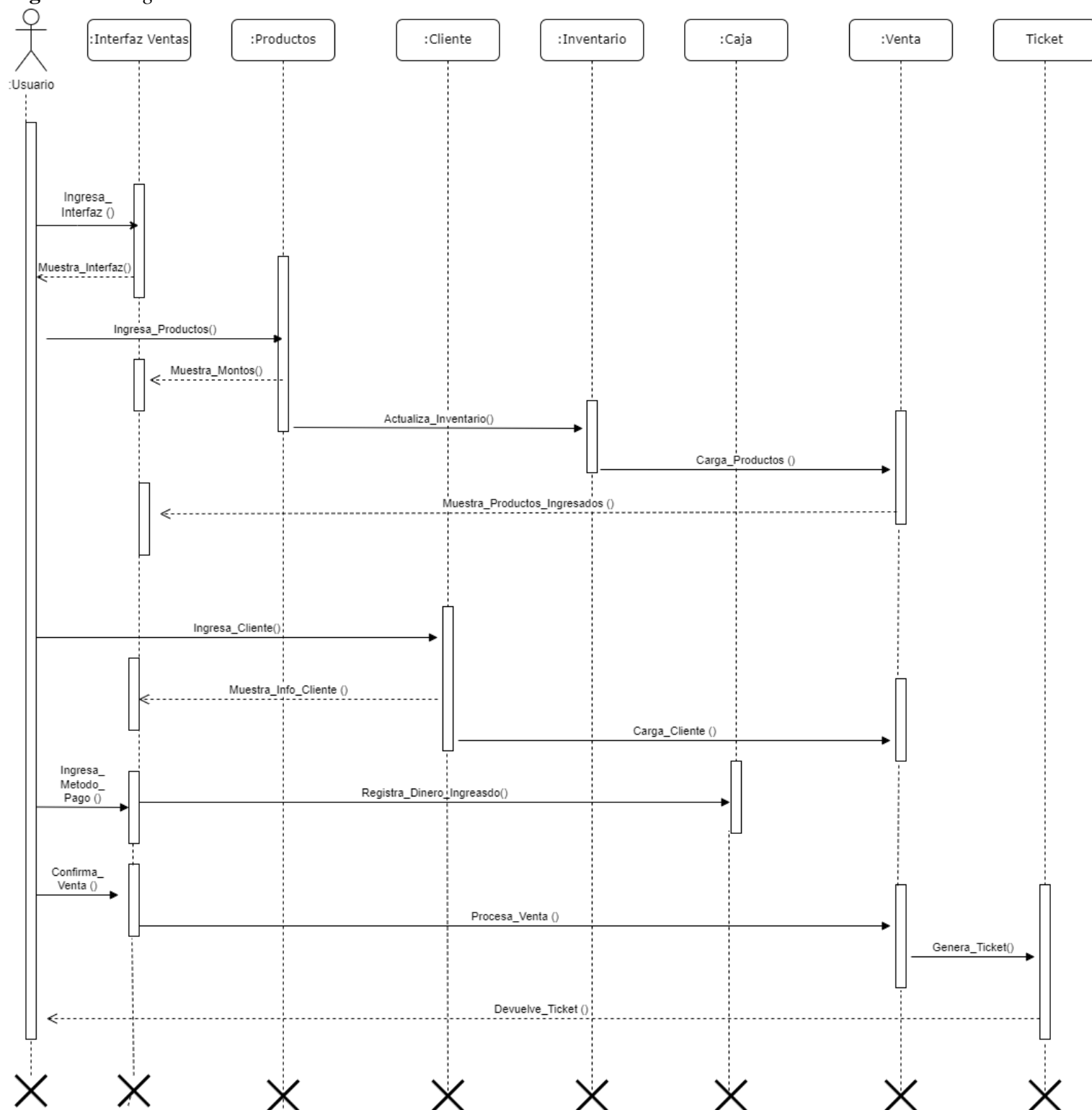
Diagramas de Secuencias

Figura 50. Diagrama de secuencias de inicio de sesión.



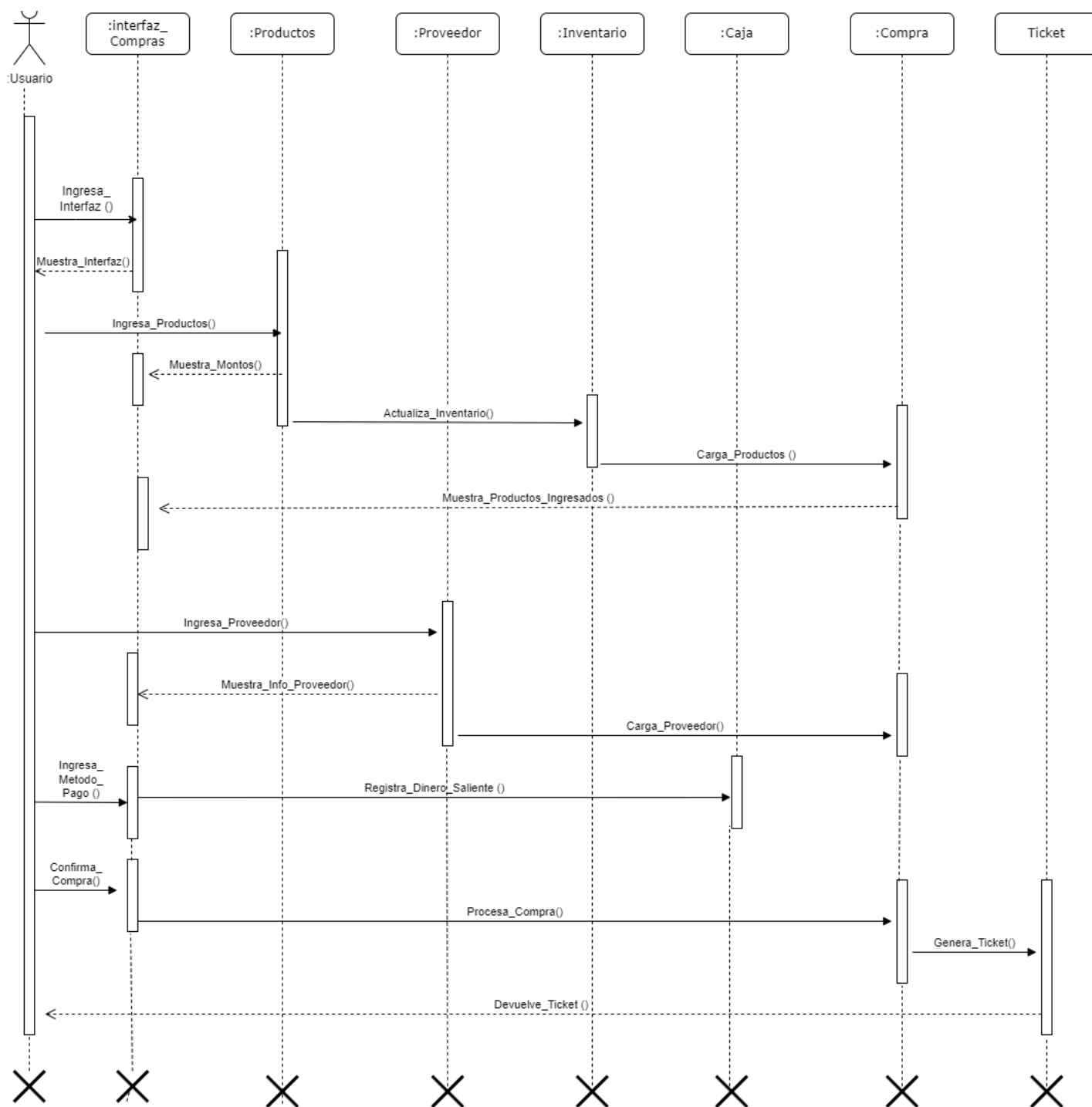
Fuente: Elaboración propia.

Figura 51. Diagrama de Secuencias de ventas.



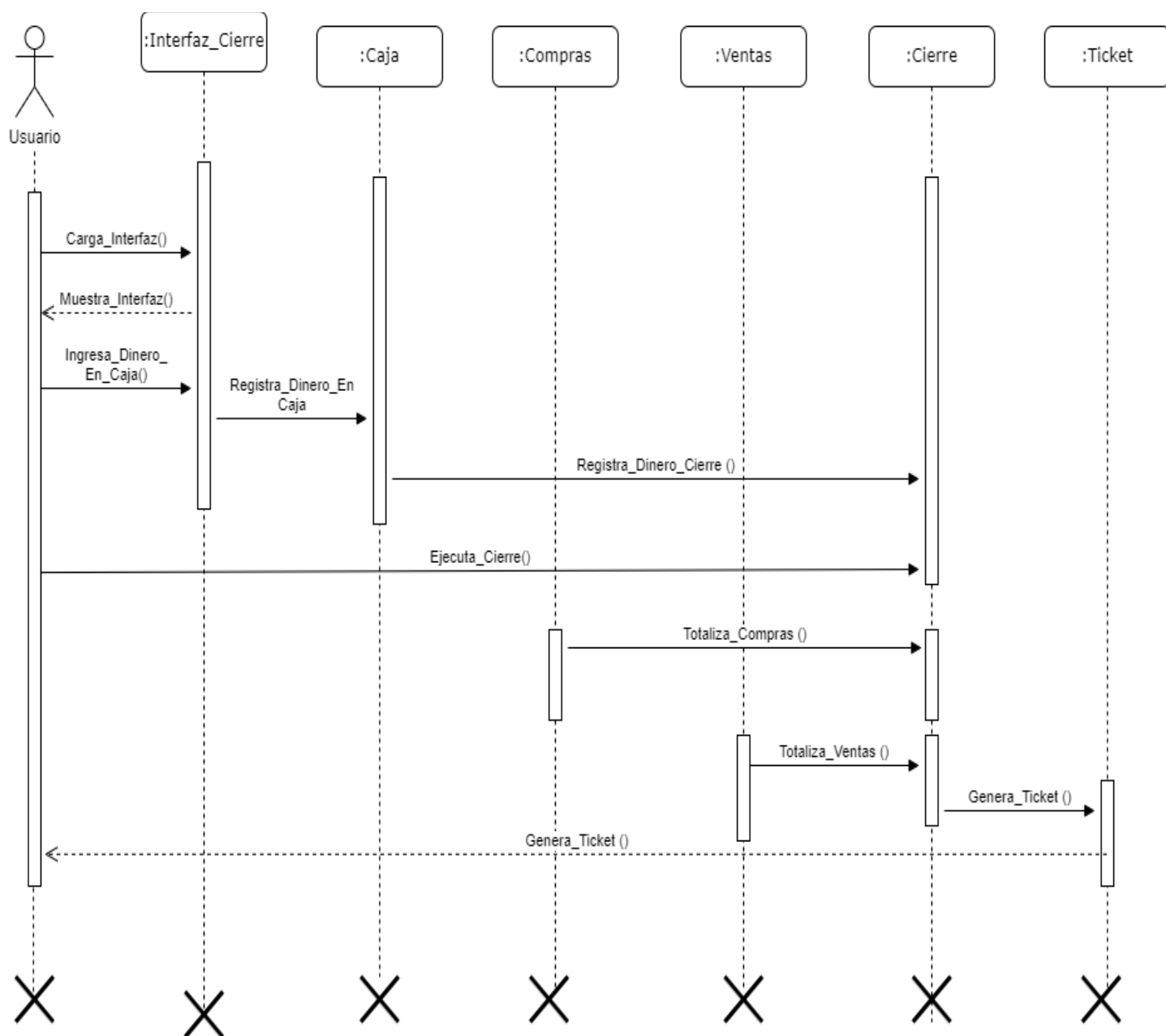
Fuente: Elaboración propia.

Figura 52. Diagrama de Secuencias de compras.



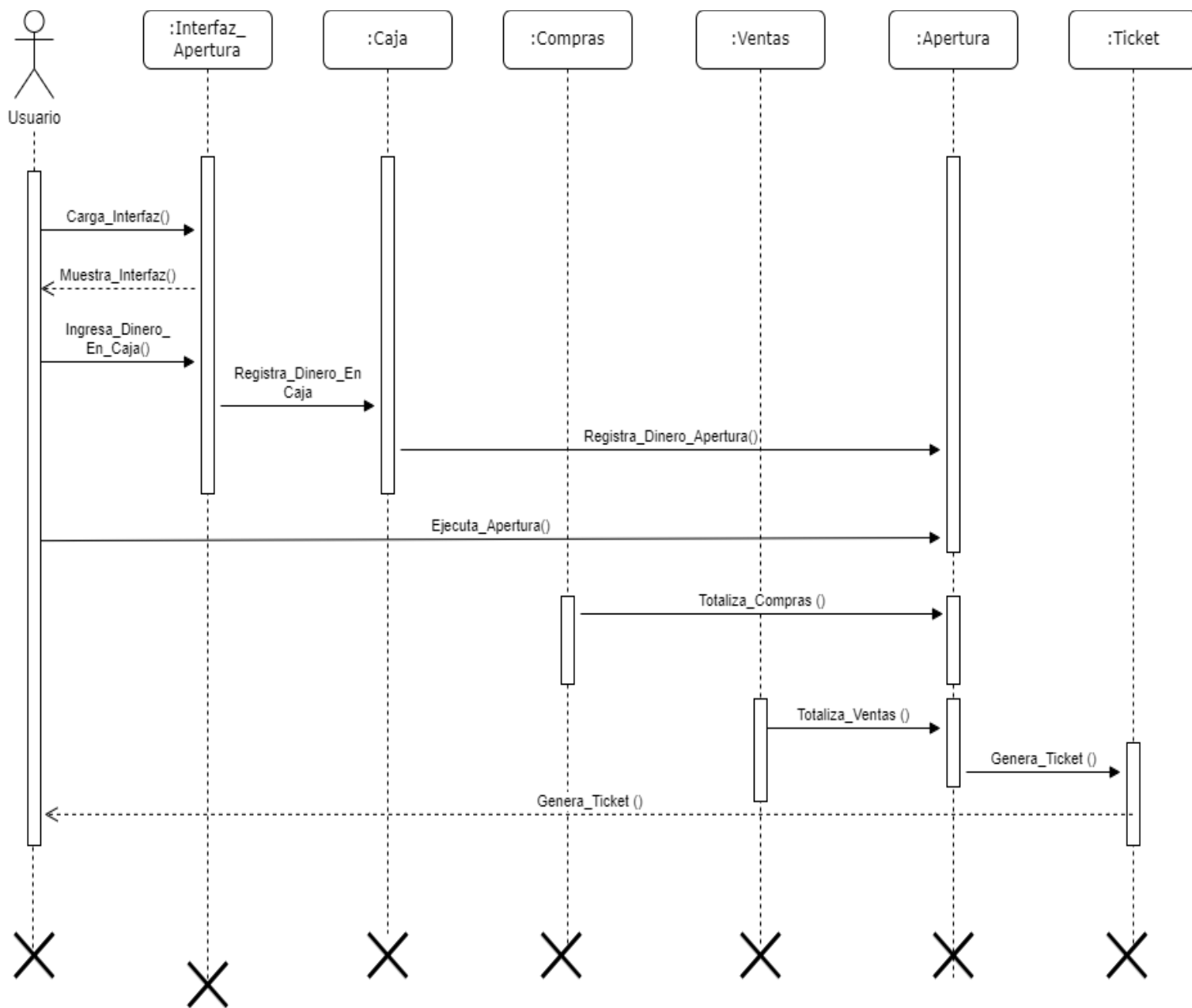
Fuente: Elaboración propia.

Figura 53. Diagrama de Secuencias de cierre de caja.



Fuente: Elaboración propia.

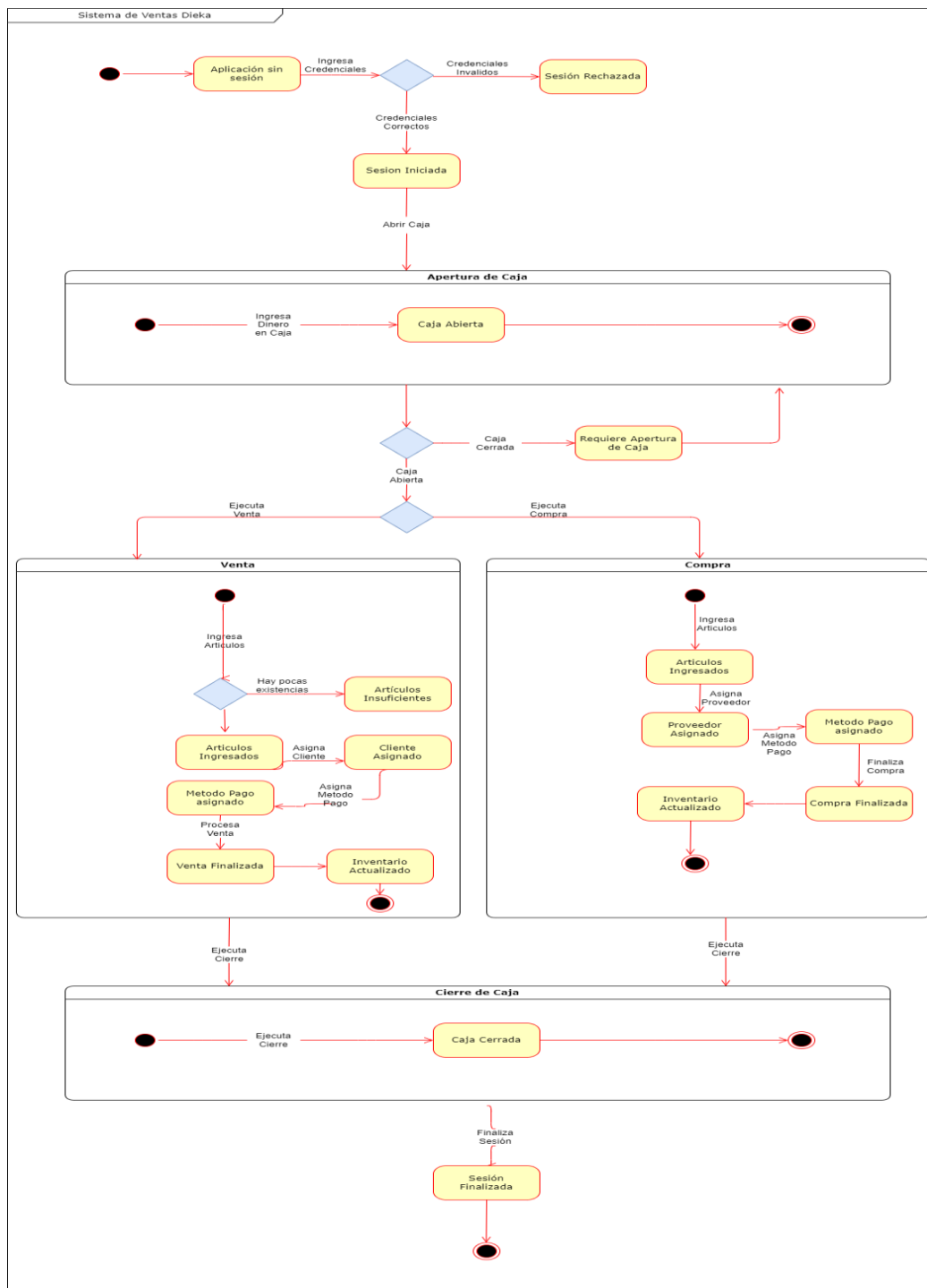
Figura 54. Diagrama de Secuencias de apertura de caja.



Fuente: Elaboración propia.

Diagramas de Máquina de Estados

Figura 55. Diagrama de máquina de estados.



Fuente: Elaboración propia.

Programación

Entradas y Salidas

En la siguiente figura se muestra un ejemplo de entrada de información para el registro de información de clientes, la entrada de información consta de dos partes el formulario en HTML que se le muestra al usuario y la clase de modelo de datos en donde se asignan las variables de lado del servidor.

Figura 56. *Diseño de entrada de registro de clientes, primera parte.*

```

<div id="collapseOne" class="collapse show" aria-labelledby="headingOne"
  data-parent="#accordionExample">
  <div class="card-body">
    <div class="form-group">
      @*Carga los campos de cedula para realizar el registro de un cliente*@
      @if (Model.Input.Cedula.Equals(0))
      {
        @*Campo para ingresar tipo de identificación*@
        <div class="form-group">
          <select id="ddlTipoCedula" class="form-control" asp-for="Input.OpcionCedula" asp-items="@Model.Input.opcionCedula">
            <option selected="true" disabled="disabled"><---Seleccione tipo de identificación---></option>
          </select>
        </div>
        @*Campo para ingresar cedula*@
        <div class="form-group" id="CampoCedula" style="display:none">
          <input asp-for="Input.Cedula" placeholder="Cedula" class="form-control" aria-describedby="CedulaAyuda" id="InputCedula" autofocus />
          <span asp-validation-for="Input.Cedula" class="text-danger"></span>
        </div>
        @*Campo para ingresar Cedula Juridica*@
        <div class="form-group" id="CampoCedulaJuridica" style="display:none">
          <input asp-for="Input.CedulaJuridica" placeholder="Cedula Juridica" class="form-control" aria-describedby="CedulaAyuda" id="InputCedulaJuridica" autofocus />
          <span asp-validation-for="Input.CedulaJuridica" class="text-danger"></span>
        </div>
        @*Campo para ingresar Cedula Residencia*@
        <div class="form-group" id="CampoCedulaResidencia" style="display:none">
          <input asp-for="Input.CedulaResidencia" placeholder="Cedula Residencia" class="form-control" aria-describedby="CedulaAyuda" id="InputCedulaResidencia" autofocus />
          <span asp-validation-for="Input.CedulaResidencia" class="text-danger"></span>
        </div>
        @*Campo para ingresar Cedula NITE*@
        <div class="form-group" id="CampoCedulaNITE" style="display:none">
          <input asp-for="Input.CedulaNITE" placeholder="Cedula NITE" class="form-control" aria-describedby="CedulaAyuda" id="InputCedulaNITE" autofocus />
          <span asp-validation-for="Input.CedulaNITE" class="text-danger"></span>
        </div>
      }
    </div>
  </div>

```

Fuente: Elaboración propia.

Figura 57. Diseño de entrada de registro de clientes, segunda parte.

```

else
{
  @*Carga los campos de cedula para realizar la actualizacion de un cliente a diferencia de los registros de cliente estos campos son readonly*@
  <input asp-for="Input.CedulaActualizar" placeholder="Cedula" class="form-control" aria-describedby="CedulaAyuda" readonly />
  @*Campo para ingresar Cedula Juridica*@
  <div class="form-group" id="CampoCedulaJuridica" style="display:none">
    <input asp-for="Input.CedulaJuridica" placeholder="Cedula Juridica" class="form-control" aria-describedby="CedulaAyuda" id="InputCedulaJuridica" autofocus />
    <span asp-validation-for="Input.CedulaJuridica" class="text-danger"></span>
  </div>
  @*Campo para ingresar Cedula Residencia*@
  <div class="form-group" id="CampoCedulaResidencia" style="display:none">
    <input asp-for="Input.CedulaResidencia" placeholder="Cedula Residencia" class="form-control" aria-describedby="CedulaAyuda" id="InputCedulaResidencia" autofocus />
    <span asp-validation-for="Input.CedulaResidencia" class="text-danger"></span>
  </div>
  @*Campo para ingresar Cedula NITE*@
  <div class="form-group" id="CampoCedulaNITE" style="display:none">
    <input asp-for="Input.CedulaNITE" placeholder="Cedula NITE" class="form-control" aria-describedby="CedulaAyuda" id="InputCedulaNITE" autofocus />
    <span asp-validation-for="Input.CedulaNITE" class="text-danger"></span>
  </div>
}
<div id="CedulaAyuda" class="form-text">
  @*<small>El numero de cédula tiene que contener 9 dígitos numéricos.</small>*@
</div>
</div>
@*Campo para ingresar el nombre del cliente*@
<div class="form-group">
  <input asp-for="Input.Nombre" placeholder="Nombre" class="form-control" />
  <span asp-validation-for="Input.Nombre" class="text-danger"></span>
</div>
@*Campo para ingresar Primer Apellido del cliente*@
<div class="form-group">
  <input asp-for="Input.Apellido1" placeholder="Primer Apellido" class="form-control" />
  <span asp-validation-for="Input.Apellido1" class="text-danger"></span>
</div>
@*Campo para ingresar Segundo Apellido del cliente*@
<div class="form-group">
  <input asp-for="Input.Apellido2" placeholder="Segundo Apellido" class="form-control" />
  <span asp-validation-for="Input.Apellido2" class="text-danger"></span>
</div>

```

Fuente: Elaboración propia.

Figura 58. Diseño de entrada de registro de clientes, tercera parte.

```

@*Campo para ingresar Numero de telefono del cliente*@
<div class="form-group">
  <input asp-for="Input.Phone" placeholder="Numero de Telefono" class="form-control" />
  <span asp-validation-for="Input.Phone" class="text-danger"></span>
</div>
@*Campo para ingresar Correo del cliente*@
<div class="form-group">
  <input asp-for="Input.Email" placeholder="Correo Electronico" class="form-control" />
  <span asp-validation-for="Input.Email" class="text-danger"></span>
</div>
@*Campo para ingresar Direccion exdacta del cliente*@
<div class="form-group">
  <input asp-for="Input.Direction" placeholder="Direccion" class="form-control" />
  <span asp-validation-for="Input.Direction" class="text-danger"></span>
</div>
@*Campo para seleccionar provincia de Direccion de domicilio del cliente*@
<div class="form-group">
  <select id="ddlProvincia" class="form-control" asp-for="InputProvincia.provincia" asp-items="@Model.InputProvincia.itemProvincia">
    <option selected="true" disabled="disabled">--Seleccionar Provincia--</option>
  </select>
  <span asp-validation-for="InputProvincia.provincia" class="text-danger"></span>
</div>
@*Campo para seleccionar provincia de Direccion de domicilio del cliente*@
<div class="form-group">
  <select id="ddlCanton" class="form-control" asp-for="InputProvincia.canton">
    <option selected="true" disabled="disabled">--Seleccionar Canton--</option>
  </select>
  <span asp-validation-for="InputProvincia.canton" class="text-danger"></span>
</div>
@*Campo para seleccionar distrito de Direccion de domicilio del cliente*@
<div class="form-group">
  <select id="ddlDistrito" class="form-control" asp-for="InputProvincia.distrito">
    <option selected="true" disabled="disabled">--Seleccionar Distrito--</option>
  </select>
  <span asp-validation-for="InputProvincia.distrito" class="text-danger"></span>
</div>

```

Fuente: Elaboración propia.

Figura 59. Diseño de entrada de registro de clientes, cuarta parte.

```

    @*Campo para seleccionar canton de Direccion de domicilio del cliente*@
    <div class="form-group">
        <div class="col-md-10">
            <input asp-for="Input.Credit" class="form-check-input" />
        </div>
        <label asp-for="Input.Credit" class="col-md-10 control-label">¿Elegible Credito?</label>
    </div>
    @*botones para enviar el formulario*@
    <div class="form-group">
        @*Si se quiere realizar un registro de un cliente el botón sera Registrar*@
        @if (Model.Input.Cedula.Equals(0))
        {
            <button type="submit" class="btn btn-primary btn-block">Registrar</button>
        }
        @*Si se quiere realizar una actualización de un cliente el botón sera Actualizar*@
        else
        {
            <button type="submit" class="btn btn-primary btn-block">Actualizar</button>
        }
    </div>
    @*Si se genera algun error en el proceso este se ira visualizar en el siguiente campo*@
    <div class="form-group">
        <label class="text-danger">@Html.DisplayFor(m => m.Input.ErrorMessage)</label>
    </div>
</div>

```

Fuente: Elaboración propia.

Como se mencionó en el párrafo anterior adicional al código del formulario los datos son almacenados en una clase de modelo para ser manipulados en los procesos de lado del servidor:

Figura 60. Clase de modelo de registro de clientes, primera parte.

```

//Propiedad que almacena el Id de Cliente
0 referencias
public int idCliente { get; set; }

//Propiedad que almacena el numero de cedula Nacional, este es un valor numero de nueve digitos según se observa en la expresion regular
[Required(ErrorMessage = "El campo Cedula Nacional es obligatorio.")]
[RegularExpression(@"^[0-9]{1})?[-. ]?[0-9]{4}[-. ]?[0-9]{4}$", ErrorMessage = "El formato de cedula nacional ingresado no es válido")]
46 referencias
public long Cedula { get; set; }

//Propiedad que almacena el numero de cedula en caso de una actualización de un registro.
1 referencias
public long CedulaActualizar { get; set; }

//Propiedad que almacena el numero de cedula juridica, este es un valor numero de diez digitos según se observa en la expresion regular
[Required(ErrorMessage = "El campo Cedula Juridica es obligatorio.")]
[RegularExpression(@"^[0-9]{1})?[-. ]?[0-9]{4}[-. ]?[0-9]{5}$", ErrorMessage = "El formato de cedula Juridica ingresado no es válido")]
7 referencias
public long CedulaJuridica { get; set; }

//Propiedad que almacena el numero de cedula DIMEX, este es un valor numero de doce digitos según se observa en la expresion regular
[Required(ErrorMessage = "El campo documento identificación Dimex es obligatorio.")]
[RegularExpression(@"^[0-9]{1})?[-. ]?[0-9]{5}[-. ]?[0-9]{6}$", ErrorMessage = "El formato de documento de identificación DIMEX ingresado no es válido")]
7 referencias
public long CedulaResidencia { get; set; }

//Propiedad que almacena el numero de cedula NITE, este es un valor numero de diez digitos según se observa en la expresion regular
[Required(ErrorMessage = "El campo documento identificación NITE es obligatorio.")]
[RegularExpression(@"^[0-9]{1})?[-. ]?[0-9]{4}[-. ]?[0-9]{5}$", ErrorMessage = "El formato de documento de identificación NITE ingresado no es válido")]
7 referencias
public long CedulaNITE { get; set; }

//Propiedad que almacena el nombre del cliente este campo es obligatorio
[Required(ErrorMessage = "El campo Nombre es obligatorio.")]
23 referencias
public string Name { get; set; }

```

Fuente: Elaboración propia.

Figura 61. Clase de modelo de registro de clientes, segunda parte.

```
//Propiedad que almacena el primer apellido del cliente este campo es obligatorio
[Required(ErrorMessage = "El campo de Primer Apellido es obligatorio.")]
20 referencias
public string Apellido1 { get; set; }

//Propiedad que almacena el segundo apellido del cliente este campo es obligatorio
[Required(ErrorMessage = "El campo de Segundo Apellido es obligatorio.")]
20 referencias
public string Apellido2 { get; set; }

//Propiedad que almacena el correo del cliente este campo es obligatorio tiene una validación de formato de correo electronico
[Required(ErrorMessage = "El campo Email es obligatorio.")]
[EmailAddress(ErrorMessage = "El campo correo electronico no es una dirección de correo electrónico válida")]
16 referencias
public string Email { get; set; }

//Propiedad que almacena el nombre del cliente este campo es obligatorio
[Required(ErrorMessage = "El campo Direccion es obligatorio.")]
9 referencias
public string Direction { get; set; }

//Propiedad que almacena el telefono del cliente este campo es obligatorio
[Required(ErrorMessage = "El campo Telefono es obligatorio.")]
[DataType(DataType.PhoneNumber)]
[RegularExpression(@"^\(?([0-9]{2})?\)?[-. ]?([0-9]{2})[-. ]?([0-9]{4})$", ErrorMessage = "El formato de telefono ingresado no es válido")]
15 referencias
public string Phone { get; set; }

//Propiedad que almacena la fecha de registro al sistema del cliente este campo es obligatorio, sin embargo el mismo se asigna en
//el proceso con la fecha actual del sistema
[Required(ErrorMessage = "El campo Fecha es obligatorio.")]
[DataType(DataType.Date)]
0 referencias
public DateTime Date { get; set; }

//Propiedad que identifica si el cliente acepta crédito o no
13 referencias
public bool Credit { get; set; }
```

Fuente: Elaboración propia.

Figura 62. Clase de modelo de registro de cliente, tercera parte.

```
//Propiedad que almacena la imagen del registro
20 referencias
public byte[] Image { get; set; }

//Propiedad que almacena el nombre de Distrito seleccionado
2 referencias
public string nombreDistrito { get; set; }

//Propiedad que almacena el nombre de Canton seleccionado
2 referencias
public string nombreCanton { get; set; }

//Propiedad que almacena el nombre de provincia seleccionado
2 referencias
public string nombreProvincia { get; set; }

//Propiedad que almacena la dirección exacta del cliente
2 referencias
public string direccionExacta { get; set; }
```

Fuente: Elaboración propia.

En el sistema se cuenta con varios tipos de salida tanto en pantalla, en formato de impresión y en formato PDF. A continuación, se puede observar un ejemplo de formato de salida en pantalla para la información detallada de un cliente:

Figura 63. Salida en pantalla de módulo de clientes, primera parte.

```

@{
}
@{
var image = "data:image/jpg;base64," + Convert.ToBase64String(Model.Input.datosCliente.Image, 0, Model.Input.datosCliente.Image.Length);
var name = Model.Input.datosCliente.Name + " " + Model.Input.datosCliente.Apellido1 + " " + Model.Input.datosCliente.Apellido2;
}

<div class="container">
<h1>@name</h1>
<div class="row">
<div class="col-sm">
<div class="card text-center" style="width: 21rem;">
<div class="card-header">
@*Muestra la imagen que tiene cargada el registro de cliente*@

</div>
<div class="card-body">
<div class="col-md-10">
<div class="row">
@*Muestra el nombre completo del cliente*@
<p>@Model.Input.datosCliente.Name</p>
</div>
</div>
@*Coloca Disponible si el cliente es elegible a credito y No Disponible si no es elegible a credito.*@
<div class="row">
<p>Credito: </p>
&nbsp;
<@if (Model.Input.datosCliente.Credito)
{
<p class="text-success">Disponible</p>
}
else
{
<p class="text-danger">No Disponible</p>
}
</div>
</div>

```

Fuente: Elaboración propia.

Figura 64. Salida en pantalla de módulo de clientes, segunda parte.

```

@*Muestra la información de credito del cliente*@
<div class="row">
<a asp-area="Clientes" asp-page="CreditoCliente" asp-route-id="@Model.Input.datosCliente.Cedula" class="btn btn-info">Credito de Cliente</a>
</div>
</div>
</div>
<div class="col-sm">
<div class="card">
<div class="card-body">
@*La siguiente sección muestra el detalle de la información del cliente*@
<table class="tableCursos">
<tbody>
<tr>
<th>
Informacion
</th>
</tr>
<tr>
<th>
@*Muestra el numero de cedula del cliente*@
Numero de Cédula
</th>
</tr>
<tr>
<td>
<p>@Model.Input.datosCliente.Cedula</p>
</td>
</tr>
<tr>
<th>
@*Muestra al Nombre del cliente*@
Nombre
</th>
</tr>
<tr>
<td>
<p>@name</p>
</td>
</tr>

```

Fuente: Elaboración propia.

Figura 65. Salida en pantalla de módulo de clientes, tercera parte.

```
@*Muestra el telefono del cliente*@
<tr>
  <th>
    Telefono
  </th>
</tr>
<tr>
  <td>
    <p>@Model.Input.datosCliente.Phone</p>
  </td>
</tr>
@*Muestra el correo electronico del cliente*@
<tr>
  <th>
    Correo Electronico
  </th>
</tr>
<tr>
  <td>
    <p>@Model.Input.datosCliente.Email</p>
  </td>
</tr>
<tr>
  @*Muestra la provincia de la dirección de domiclilio del cliente*@
  <th>
    Provincia
  </th>
</tr>
<tr>
  <td>
    <p>@Model.Input.datosCliente.nombreProvincia</p>
  </td>
</tr>
@*Muestra el canton de la dirección de domiclilio del cliente*@
<tr>
  <th>
    Canton
  </th>
</tr>
<tr>
```

Fuente: Elaboración propia.

Figura 66. Salida en pantalla de módulo de clientes, cuarta parte.

```

    /*Muestra el distrito de la dirección de domicilio del cliente*/
    <tr>
        <th>
            Distrito
        </th>
    </tr>
    <tr>
        <td>
            <p>@Model.Input.datosCliente.nombreDistrito</p>
        </td>
    </tr>
    /*Muestra la dirección exacta de domicilio del cliente*/
    <tr>
        <th>
            Dirección Exacta
        </th>
    </tr>
    <tr>
        <td>
            <p>@Model.Input.datosCliente.direccionExacta</p>
        </td>
    </tr>
</td>

```

Fuente: Elaboración propia.

Los campos observados anteriormente en el HTML se cargan de información en el método OnGet() de la vista, según se observa a continuación:

Figura 67. Clase de salida en pantalla, primera parte.

```

namespace FactuLib.Areas.Clientes.Pages.Account
{
    [Authorize]
    6 referencias
    public class DetailsModel : PageModel
    {
        // Declara propiedad para el manejo de los metodos de la clase cliente
        private ICliente _clientes;

        // Metodo constructor para declarar propiedades
        0 referencias
        public DetailsModel(
            ApplicationDbContext context)
        {
            _clientes = new ICliente(context);
        }

        //Metodo On Get que se ejecuta al cargar la vista
        0 referencias
        public void OnGet(long id)
        {
            //Trae los datos del cliente según el metodo getTClientesAsync
            //de la clase de clientes y los ingresa en una variable
            var data = _clientes.getTClienteAsync(null, id);

            // Si el metodo devuelve datos carga datos al
            // formulario utilizando la clase de modelo
            if (data.Count > 0)
            {
                Input = new InputModel
                {
                    datosCliente = data.ToList().Last()
                };
            }
        }
    }
}

```

Fuente: Elaboración propia.

Figura 68. Clase de salida en pantalla, segunda parte.

```

// Propiedad de la clase de modelo

[BindProperty]
19 referencias
public InputModel Input { get; set; }

//Clase de Modelo para almacenar la información
2 referencias
public class InputModel
{
    19 referencias
    public InputModelRegistrar datosCliente { get; set; }
}

```

Fuente: Elaboración propia.

Adicional, para la extracción de reportes se utiliza la librería de Rotativa para generar vistas HTML en formato PDF, el código a continuación muestra la forma en que son generados los documentos PDF:

Figura 69. Clase de generación de PDF.

```

0 referencias
public ActionResult ReportesClientes(String opcion)
{
    Object[] objects = new Object[3];
    // Realiza la consulta de clientes y lo almacena en una variable
    var data = _cliente.getIClienteAsync(null, 0);
    var url = Request.Scheme + "://" + Request.Host.Value;
    objects = new LPaginador<InputModelRegistrar>().paginador(data, 0, 10, "Clientes", "Clientes", "Reportes", url);
    // Define la URL de la Cabecera
    string _headerUrl = Url.Action("HeaderPDF", "Clientes", null, "https");
    // Define la URL del Pie de página
    string _footerUrl = Url.Action("FooterPDF", "Clientes", null, "https");

    // Define la clase de modelo para enviar a la vista HTML que se convertira en PDF
    models = new DataPaginador<InputModelRegistrar>
    {
        List = (List<InputModelRegistrar>)objects[2],
        Pagi_info = (String)objects[0],
        Pagi_navegacion = (String)objects[1],
        Input = new InputModelRegistrar()
    };

    //Genera PDF a partir de la clase ViewAsPDF de la libreria de Rotativa
    return new ViewAsPdf("Reports/ReportesClientes", models)
    {
        //..
    };
}
0 referencias

```

Fuente: Elaboración propia.

Adicional al tipo de salida en formato PDF, para ciertos módulos como lo son compras y ventas se utilizan salidas en formato de impresión, los mismos son construidos

mediante la clase LTicket del prototipo, donde esta clase utiliza los métodos de la clase “StringBuilder” para construir el texto y enviarlo a una clase con los métodos necesarios para generar la impresión:

Figura 70. *Formato de salida de Ticket.*

```

2 referencias
public void GeneraTicket(int numeroTicket, string nombreCliente, string userName, List<TemporalVentas> detalles, float montoBruto, float montoIVA, float montoDesc, float montoNeto)
{
    //Genera el ticket que se le entrega al cliente al momento de la venta.
    LTicket TicketPago = new LTicket();
    //TicketPago.AbrirConexion();
    TicketPago.TextoCentro("Librería y Bazar Dieka");
    TicketPago.TextoIzquierda("Dirección");
    TicketPago.TextoIzquierda("La Trinidad de Moravia");
    TicketPago.TextoIzquierda("Telefono: 22299553");
    TicketPago.LineasGuion();
    TicketPago.TextoCentro("¡Tiquete de Venta!");
    TicketPago.LineasGuion();
    TicketPago.TextoIzquierda($"Tiquete#: {numeroTicket}");
    TicketPago.TextoIzquierda($"Cliente: {nombreCliente}");
    TicketPago.TextoIzquierda($"Fecha: {DateTime.Now}");
    TicketPago.TextoIzquierda($"Usuario: {userName}");
    TicketPago.LineaAsteriscos();
    TicketPago.TextoCentro("Productos Vendidos");
    TicketPago.TextoExtremo("NombreProducto", "Cantidad/Monto");
    foreach (var item in detalles)
    {
        TicketPago.TextoExtremo($"item.TProducto.Nombre_Producto", $"Cant {item.Cantidad_Venta} ${String.Format("{0:#,###,###,##0.00###}", item.TotalNeto)}");
    }
    TicketPago.LineaAsteriscos();
    TicketPago.TextoDerecha($"Bruto: {String.Format("{0:#,###,###,##0.00###}", montoBruto)}");
    TicketPago.TextoDerecha($"I.V.A: {String.Format("{0:#,###,###,##0.00###}", montoIVA)}");
    TicketPago.TextoDerecha($"Desc: {String.Format("{0:#,###,###,##0.00###}", montoDesc)}");
    TicketPago.TextoDerecha($"Neto: {String.Format("{0:#,###,###,##0.00###}", montoNeto)}");
    TicketPago.LineaAsteriscos();
    TicketPago.TextoCentro("Muchas gracias");
    //TicketPago.CortaTicket();
    TicketPago.ImprimirTicket("Microsoft XPS Document Writer");
}

```

Fuente: Elaboración propia.

Procesos

A continuación, se presentan ejemplos de dos de los procesos más importantes presentes en el prototipo estos son el proceso de ventas y el proceso de cierre de ventas.

Figura 71. *Proceso de venta, primera parte.*

```

1 referencia
private async Task<bool> VentasAsync()
{
    _dataInput = Input; // Variable que almacena los datos que fueron ingresados en el formulario
    var valor = false; // Variable utilizada para dar el resultado de la operación satisfactorio o fallido.
    var idUser = userManager.GetUserId(User); // Devuelve el Id del usuario logueado en el sistema
    var userReg = _context.TUsers.Where(u => u.IdUser.Equals(idUser)).ToList().Last(); // Devuelve el registro de usuario logueado al sistema
    var cliente = _context.TClientes.Where(u => u.Nombre.Replace(" ", "").Equals(_clienteSeleccionado.Replace(" ", ""))).ToList().Last(); // Retorna el registro de cliente de la venta
    var productos = _context.TProducto.ToList(); // Retorna el listado de productos registrados en el sistema.
    var clientes = _context.TClientes.ToList(); //Retorna el listado de cliente registrados en el sistema.
    var creditosCliente = _context.TCreditosClientes.ToList(); // Retorna el listado de creditos de clientes registrados en el sistema
    var credito = creditosCliente.Where(c => c.TClientes.IdCliente.Equals(cliente.IdCliente)).ToList().Last(); // Retorna el credito del cliente de la venta
    var RegistrosVentas = _context.TRegistroVentas.ToList(); // Retorna los registros de ventas en el sistema
    int numeroTicket = 0; // Numero de ticket asignado a la venta

    // Si no existen ventas registradas el numero de ticket sera 1, de lo contrario el numero de ticket sera el ultimo registro +1.
    if (RegistrosVentas.Count == 0)
    {
        numeroTicket = 1;
    }
    else
    {
        numeroTicket = RegistrosVentas.ToList().Last().Id_RegistroVentas + 1;
    }

    //Trae los datos de la tabla temporal de ventas donde contiene los productos a vender.
    List<TemporalVentas> datosTemporal = _context.TTemporalVentas.Where(u => u.TUser.IdUser.Equals(idUser) && u.TCliente.IdCliente.Equals(cliente.IdCliente)).ToList();
}

```

Fuente: Elaboración propia.

Figura 72. Proceso de venta, segunda parte.

```

if (datosTemporal.Count > 0)
{
    var strategy = _context.Database.CreateExecutionStrategy(); // Apertura una estrategia de base de datos para abrir una transaccion.
    await strategy.ExecuteAsync(async () =>
    {
        using (var transaction = _context.Database.BeginTransaction()) // Abre una transacción de base de datos
        {
            try
            {
                string _ticket = null; // Declara variable de ticket
                var dateNow = DateTime.Now; // Declara variable con fecha actual del sistema
                var metodoPago = 0; // Declara variable de metodo de pago
                var _cambio = 0.0m; // Declara la variable de cambio al momento de pagar la venta.
                var user = _context.TUsers.Where(u => u.IdUser.Equals(idUser)).ToList(); // Declara la variable de usuario
                var nameUser = $"{user[0].Name} {user[0].Apellido1} {user[0].Apellido2}"; // Declara la variable de nombre de usuario
                var nameCliente = $"{cliente.Nombre} {cliente.Apellido1} {cliente.Apellido2}"; // Declara el nombre del cliente
                var monto = _ventas.getMontoTotal(); //Calcula el monto neto de la venta
                var montoBruto = _ventas.getMontoBruto(); // Calcula el monto bruto de la venta
                var montoDescuento = _ventas.getMontoDescuentos(); // Calcula el monto descuento de la venta
                var montoImpuesto = _ventas.getMontoImpuestos(); // Calcula el monto de impuesto de la venta
                var dataTempo = datosTemporal.Last(); // declara variable con el ultimo registro de la tabla temporal
                var deuda = monto - Input.Pagos; // Declara el monto de la deuda de la venta en caso de ser a credito.

                // Si la venta es a contado realiza los siguientes procedimientos
                if (_dataInput.Contado == true)
                {
                    //Establece el metodo de pago
                    if (_dataInput.ChkEfectivo == true)
                    {
                        metodoPago = 1;
                    }
                    if (_dataInput.ChkTarjeta == true)
                    {
                        metodoPago = 2;
                    }
                    if (_dataInput.ChkTransferencia == true)
                    {
                        metodoPago = 3;
                    }
                }
            }
        }
    });
}

```

Fuente: Elaboración propia.

Figura 73. Proceso de venta, tercera parte.

```

//Calcula el cambio del sistema
_cambio = (decimal)(_dataInput.Pagos - monto);

//Declara el objeto de Registro de venta a cargarse en base de datos
var RegistroVenta = new TRegistroVentas
{
    Total_Bruto = montoBruto,
    Total_Descuento = montoDescuento,
    Total_IVA = montoImpuesto,
    Total_Neto = monto,
    MetodoPago = metodoPago,
    DineroRecibido = _dataInput.Pagos,
    CambioCompra = _cambio,
    Fecha_Compra = DateTime.Now,
    TClientes = cliente,
    TUser = userReg,
    Estado_Venta = true
};

//Agrega el objeto en base de datos
await _context.AddAsync(RegistroVenta);
// Guarda cambios
await _context.SaveChangesAsync();

//Agrega todos los productos ingresados en la tabla temporal de ventas, a la venta
foreach (var item in datosTemporal)
{
    var ProductoDetalle = productos.Where(p => p.Id_Producto.Equals(item.TProducto.Id_Producto)).ToList().Last();
    ProductoDetalle.Cantidad_Producto -= ProductoDetalle.Cantidad_Producto - item.Cantidad_Venta;

    //Declara el objeto de detalles de la venta
    var DetalleVenta = new TDetallesVentas
    {
        Cantidad_Producto = item.Cantidad_Venta,
        Monto_Bruto_Detalle = item.TotalBruto,
        Descuento_Detalle = item.TotalDescuentos,
        Monto_Neto_Detalle = item.TotalNeto,
        Monto_Impuesto_Detalle = item.TotalImpuestos,
        TRegistroVentas = RegistroVenta,
        TProducto = ProductoDetalle
    };
}

```

Fuente: Elaboración propia.

Figura 74. Proceso de venta, cuarta parte.

```

//Agrega el objeto en base de datos
await _context.AddAsync(DetalleVenta);
//Guarda cambios
await _context.SaveChangesAsync();

//Elimina información de la tabla temporal
_context.TTemporalVentas.Where(t => t.idTempVentas.Equals(item.idTempVentas)).ToList().ForEach(d => _context.TTemporalVentas.Remove(d));
await _context.SaveChangesAsync();
}
valor = true;
//Genera ticket al finalizar la venta
GeneraTicket(numeroTicket, nameCliente, nameUser, datosTemporal, montoBruto, montoImpuesto, montoDescuento, monto);
//Realiza commit en la transacción para guardar los registros en base de datos
transaction.Commit();
ventaIniciada = false;
}

// Si la venta es a credito realiza los siguientes procedimientos
if (_dataInput.Credito == true)
{
    //Declara los valores de deuda a ingresar para la venta

    var deudaActualAnterior = (float)credito.DeudaActual;
    var deudaNueva = deudaActualAnterior + monto;
    var deudaActualNueva = deudaNueva;

    //Declara el objeto de credito de cliente
    var CreditoCliente = new TCreditoClientes
    {
        //IdDeuda = credito.idDeuda,
        Deuda = (Decimal)deudaNueva,
        Mensual = credito.Mensual,
        Cambio = credito.Cambio,
        UltimoPago = credito.UltimoPago,
        FechaPago = credito.FechaPago,
        DeudaActual = (Decimal)deudaActualNueva,
        FechaDeuda = credito.FechaDeuda,
        Ticket = credito.Ticket,
        FechaLimite = credito.FechaLimite,
        TClientes = cliente,
    };
}

```

Fuente: Elaboración propia.

Figura 75. Proceso de venta, quinta parte.

```

//Envia a actualizar el objeto en base de datos
_context.Update(CreditoCliente);
//Guarda cambios realizados en base de datos
_context.SaveChangesAsync();

//Declara el objeto de Registro de venta a cargarse en base de datos
var RegistroVenta = new TRegistroVentas
{
    Total_Bruto = montoBruto,
    Total_Descuento = montoDescuento,
    Total_IVA = montoImpuesto,
    Total_Neto = monto,
    MetodoPago = 0,
    DineroRecibido = 0,
    CambioCompra = 0,
    Fecha_Compra = DateTime.Now,
    TClientes = cliente,
    TUser = userReg,
    Estado_Venta = true
};

//Agrega el objeto en base de datos
await _context.AddAsync(RegistroVenta);

//Guarda cambios en base de datos
await _context.SaveChangesAsync();

//Agrega todos los productos ingresados en la tabla temporal de ventas, a la venta
foreach (var item in datosTemporal)
{
    var ProductoDetalle = productos.Where(p => p.Id_Producto.Equals(item.TProducto.Id_Producto)).ToList().Last();
    ProductoDetalle.Cantidad_Producto -= ProductoDetalle.Cantidad_Producto - item.Cantidad_Venta;

    //Declara el objeto de detalles de la venta
    var DetallesVenta = new TDetallesVentas
    {
        Cantidad_Producto = item.Cantidad_Venta,
        Monto_Bruto_Detalle = item.TotalBruto,
        Descuento_Detalle = item.TotalDescuentos,
        Monto_Neto_Detalle = item.TotalNeto,
        Monto_Impuesto_Detalle = item.TotalImpuestos,
        TRegistroVentas = RegistroVenta,
        TProducto = ProductoDetalle
    };
}

```

Fuente: Elaboración propia.

Figura 76. Proceso de venta, sexta parte.

```

        // Agregue el detalle de venta a la base de datos
        await _context.AddAsync(DetallesVenta);
        // Guarda cambios
        await _context.SaveChangesAsync();

        //Elimina información de la tabla temporal
        _context.TTemporalVentas.Where(t => t.idTempVentas.Equals(item.idTempVentas)).ToList().ForEach(d => _context.TTemporalVentas.Remove(d));
        await _context.SaveChangesAsync();
    }
    valor = true;
    //Genera ticket
    GeneraTicket(numeroTicket, nameCliente, nameUser, datosTemporal, montoBruto, montoImpuesto, montoDescuento, monto);
    //Realiza commit de la transaccion para guardar registros en base de datos
    transaction.Commit();
    ventaIniciada = false;
}

//Validacion de que ingrese opciones de credito o contado
if (!_dataInput.Credito == false && !_dataInput.Contado == false)
{
    _dataInput.ErrorMessage = "Debe seleccionar una forma de Pago";
    valor = false;
}
}
catch (Exception ex)
{
    //En caso de que se genere una excepcion la carga en la propiedad error message del modelo de datos.
    _dataInput.ErrorMessage = ex.Message;
    valor = false;

    //Realiza rollback de la transaccion
    transaction.Rollback();
}
});
}
else
{
    _dataInput.ErrorMessage = "No hay productos registrados";
}
return valor;
}
}

```

Fuente: Elaboración propia.

A continuación, se muestra el proceso de cierre de caja el cual consiste en totalizar todas las operaciones de compra y venta realizadas durante el día:

Figura 77. Proceso de cierre, primera parte.

```

1 referencia
private async Task<bool> SaveAsync()
{
    _dataInput = Input; // Variable que almacena los datos que fueron ingresados en el formulario
    var result = false; //Variable que almacena el resultado de la transacción si fue satisfactoria o fallida.
    //Variable para almacenar el registro de usuario logueado al sistema
    var usuario = _context2.TUsers.Where(u => u.IdUser.Equals(_userManager.GetUserId(User))).ToList().Last();
    //Variable para almacenar los registros de venta sin estado de cierre y que realizo el usuario logueado al sistema.
    var ventas = _context2.TRegistroVentas.Where(v => v.Estado_Venta == true && v.TUser.IdUser.Equals(usuario.IdUser)).ToList();
    //Variable para almacenar los registros de compra sin estado de cierre y que realizo el usuario logueado al sistema.
    var compras = _context2.TRegistroCompras.Where(c => c.Estado_Compra == true && c.TUser.IdUser.Equals(usuario.IdUser)).ToList();
    // Variable para almacenar los registros de apertura sin estado de cierre y que realizo el usuario logueado al sistema.
    var apertura = _context2.TRegistroApertura.Where(a => a.Estado == true && a.TUser.IdUser.Equals(usuario.IdUser)).ToList();
    // Declaración de estrategia de base de datos para establecer una transacción
    var strategy = _context2.Database.CreateExecutionStrategy();

    //Variables para totalización de ventas y compras
    _dataInput.TotalVentas = _ventas.getTotalVentasCajasCierre(usuario.IdUser);
    _dataInput.TotalVentasCuentasBancarias = _ventas.getTotalVentasCuentasCierre(usuario.IdUser);
    _dataInput.TotalVentasCredito = _ventas.getTotalVentasCreditoCierre(usuario.IdUser);
    _dataInput.TotalCompras = _compras.getTotalComprasCajasCierre(usuario.IdUser);
    _dataInput.TotalComprasCuentasBancarias = _compras.getTotalComprasCuentasCierre(usuario.IdUser);
    _dataInput.TotalComprasCredito = _compras.getTotalComprasCreditoCierre(usuario.IdUser);
    _dataInput.MontoCajasApertura = _apertura.getTotalCajasApertura(usuario.IdUser);
    _dataInput.MontoCuentasApertura = _apertura.getTotalCuentasApertura(usuario.IdUser);

    //Realiza validación si existe una apertura de caja sin cierre pendiente en el sistema
    if (!_apertura.ValidaApertura(usuario.IdUser))
    {
        //Valida que los datos ingresados en el formulario sean validos
        if (ModelState.IsValid)
        {

```

Fuente: Elaboración propia.

Figura 78. *Proceso de cierre, segunda parte.*

```

await strategy.ExecuteAsync(async () =>
{
    //Comienza una transacción de base de datos
    using (var transaction = _context.Database.BeginTransaction())
    {
        try
        {
            //Variable que calcula el monto de dinero en cajas según los datos ingresados de cantidad de monedas y billetes
            var montoCajas = (_dataInput.CantMonedasCinco * 5) + (_dataInput.CantMonedasDiez * 10) + (_dataInput.CantMonedasVeinticinco * 25) + (_dataInput.CantMonedasCincuenta * 50) +
                (_dataInput.CantMonedasCien * 100) + (_dataInput.CantMonedasQuinientos * 500) + (_dataInput.CantBilletesMil * 1000) + (_dataInput.CantBilletesDosMil * 2000) +
                (_dataInput.CantBilletesCincoMil * 5000) + (_dataInput.CantBilletesDiezMil * 10000) + (_dataInput.CantBilletesVeinteMil * 20000) +
                (_dataInput.CantBilletesCincuentaMil * 50000);

            //Variable de cierre
            var cierre = new IRegistroCierre()
            {
                TUser = usuario,
                TRegistroApertura = apertura.Last(),
                Fecha_Cierre = DateTime.Now,
                Dinero_Cajas = montoCajas,
                Dinero_Cuentas = _dataInput.Dinero_Cuentas,
                Total_Ventas_Efectivo = _dataInput.TotalVentas,
                Total_Ventas_Cuentas = _dataInput.TotalVentasCuentasBancarias,
                Total_Ventas_Credito = _dataInput.TotalVentasCredito,
                Total_Compras_Efectivo = _dataInput.TotalCompras,
                Total_Compras_Credito = _dataInput.TotalComprasCredito,
                Total_Compras_Cuentas = _dataInput.TotalComprasCuentasBancarias,
                Monto_Faltante = 0,
            };

            //Guarda el cierre en base de datos
            await _context.AddAsync(cierre);

            //Guarda los cambios en base de datos
            _context.SaveChanges();

            //Actualiza los registros de ventas colocandolos como cerrados
            foreach (var item in ventas)
            {
                item.Estado_Venta = false;
                _context.Update(item);
                _context.SaveChanges();
            }
        }
    }
}

```

Fuente: Elaboración propia.

Figura 79. *Proceso de cierre, tercera parte.*

```

//Actualiza los registros de compras colocandolos como cerrados
foreach (var item in compras)
{
    item.Estado_Compra = false;
    _context.Update(item);
    _context.SaveChanges();
}

//Actualiza el registro de apertura colocandolo como cerrado
foreach (var item in apertura)
{
    item.Estado = false;
    _context.Update(item);
    _context.SaveChanges();
}

//Realiza commit de la transacción guardando los registros en base de datos
transaction.Commit();
_dataInput = null;
result = true;
}
catch (Exception ex)
{
    //Establece el mensaje de error en caso de producirse una excepción
    _dataInput.ErrorMessage = ex.Message;
    transaction.Rollback();
    result = false;
}
});
}
else
{
    foreach (var modelState in ModelState.Values)
    {
        foreach (var error in modelState.Errors)
        {
            _dataInput.ErrorMessage += error.ErrorMessage;
        }
    }
    result = false;
}
}
}

```

Fuente: Elaboración propia.

Validaciones:

Los siguientes ejemplos son algunos de las validaciones que se realizan a nivel de código fuente con el fin de velar con la veracidad y validez de los datos. A continuación, se muestra una validación que se aplica para todos los módulos y que asegura que los datos ingresados por medio del formulario cumplan el formato correcto:

Figura 80. *Validación de formatos.*

```
if (ModelState.IsValid)
{
```

Fuente: Elaboración propia.

La siguiente validación se observa en distintos segmentos del código y asegura que las operaciones se realicen sobre objetos que contengan registros y de esta forma evitar posibles problemas de nulo:

Figura 81. *Validación de existencia de datos.*

```
if (!listTClientes.Count.Equals(0))
{
```

Fuente: Elaboración propia.

La siguiente validación asegura que solo un usuario autorizado pueda realizar ciertas operaciones:

Figura 82. *Validación de rol*

```
if (User.IsInRole("Administrador"))
{
```

Fuente: Elaboración propia.

Por último, existen varias validaciones en los distintos módulos que aseguran que no existan duplicidad de registros.

Figura 83. *Validación de registros duplicados.*

```
if (!_cliente.ExisteCliente(Input.Cedula))
{
```

Fuente: Elaboración propia.

Módulos

Dentro de la arquitectura de Modelo vista controlador del lenguaje asp.NET, existe el término de áreas estas permiten separar lógicamente vistas y procedimientos según la naturaleza que cuente dentro del negocio. En la siguiente imagen se puede observar el archivo “startup.cs” en donde se visualiza un área dedicada para cada módulo.

Figura 84. Lista de áreas del proyecto.

```
endpoints.MapAreaControllerRoute("Users", "Users", "{controller=Users}/{action=Users}/{id?}");
endpoints.MapAreaControllerRoute("Principal", "Principal", "{controller=Principal}/{action=Principal}/{id?}");
endpoints.MapAreaControllerRoute("Clientes", "Clientes", "{controller=Clientes}/{action=Clientes}/{id?}");
endpoints.MapAreaControllerRoute("Proveedores", "Proveedores", "{controller=Proveedores}/{action=Proveedores}/{id?}");
endpoints.MapAreaControllerRoute("Compras", "Compras", "{controller=Compras}/{action=Compras}/{id?}");
endpoints.MapAreaControllerRoute("Productos", "Productos", "{controller=Productos}/{action=Productos}/{id?}");
endpoints.MapAreaControllerRoute("Ventas", "Ventas", "{controller=Ventas}/{action=Ventas}/{id?}");
endpoints.MapAreaControllerRoute("Cajas", "Cajas", "{controller=Cajas}/{action=Cajas}/{id?}");
endpoints.MapAreaControllerRoute("Análisis", "Análisis", "{controller=Análisis}/{action=Análisis}/{id?}");
endpoints.MapRazorPages();
```

Fuente: Elaboración propia.

De igual forma se configura un “dataset” para cada tabla diseñada en la base de datos:

Figura 85. Listado de "Data sets, primera parte

```
26 referencias
public DbSet<TUser> TUsers { get; set; }

23 referencias
public DbSet<TClientes> TClientes { get; set; }

6 referencias
public DbSet <THistoricoPagosClientes> THistoricoPagosClientes { get; set; }

11 referencias
public DbSet <TProvincia> TProvincia { get; set; }

10 referencias
public DbSet <TCanton> TCanton { get; set; }

6 referencias
public DbSet <TDistrito> TDistrito { get; set; }

3 referencias
public DbSet<TDireccionCliente> TDireccionCliente { get; set; }

6 referencias
public DbSet <TCorreosClientes> TCorreosClientes { get; set; }

3 referencias
public DbSet <TTelefonoCliente> TTelefonoCliente {get; set; }

6 referencias
public DbSet <TCreditoClientes> TCreditoClientes { get; set; }

21 referencias
public DbSet <TProveedor> TProveedor { get; set; }

4 referencias
public DbSet <TCreditoProveedor> TCreditoProveedor { get; set; }

4 referencias
public DbSet <TCorreosProveedor> TCorreosProveedor { get; set; }

4 referencias
public DbSet <TTelefonosProveedor> TTelefonosProveedor { get; set; }

5 referencias
public DbSet <TDireccionesProveedor> TDireccionesProveedor {get; set; }

4 referencias
public DbSet <THistoricoPagosProveedor> THistoricoPagosProveedor { get; set; }
```

Fuente: Elaboración propia.

Figura 86. Listado de "Data Set"

```
16 referencias
public DbSet <TTemporalCompras> TTemporalCompras { get; set; }

9 referencias
public DbSet <TRegistroCompras> TRegistroCompras { get; set; }

1 referencia
public DbSet <TDetallesCompras> TDetallesCompras { get; set; }

16 referencias
public DbSet<TTemporalVentas> TTemporalVentas { get; set; }

10 referencias
public DbSet<TRegistroVentas> TRegistroVentas { get; set; }

1 referencia
public DbSet<TDetallesVentas> TDetallesVentas { get; set; }

6 referencias
public DbSet <TTipoProducto> TTipoProducto { get; set; }

29 referencias
public DbSet <TProducto> TProducto { get; set; }

4 referencias
public DbSet <TRegistroApertura> TRegisroApertura { get; set; }

0 referencias
public DbSet <TRegistroCierre> TRegisroCierre { get; set; }
```

Fuente: Elaboración propia.

Pruebas

Tabla 63. *Caso de prueba de inicio y cierre de sesión.*

Caso de Prueba de inicio y cierre de sesión	<Código del CP>	CP-01
<p>Descripción: En el siguiente caso de prueba se busca realizar el proceso completo de inicio de sesión al sistema, permitiendo así el acceso a un usuario autorizado.</p>		
<p>Prerrequisitos -El usuario tiene que contar con credenciales previamente creados en el sistema. -El usuario tiene que conocer su nombre de usuario y contraseña.</p>		
<p>Pasos: Seguir los pasos enumerados en el caso de uso CU-01, Ingreso al sistema. Seguir los pasos enumerados en el caso de uso CU-02, Salida del sistema.</p>		
<p>Resultado esperado: El usuario debe poder ingresar al sistema utilizando las credenciales configuradas en el sistema y otorgados previamente, de igual forma tiene que ser capaz de cerrar la sesión al finalizar de utilizar el sistema.</p>		
<p>Resultado obtenido: El usuario realiza la prueba correspondiente utilizando el correo dmendezrod.12@gmail.com y la contraseña correspondiente para este usuario, dicho acceso fue previamente registrado en el sistema. Al colocar los accesos en la página de inicio de sesión y dar el botón de ingresar se obtuvo el resultado deseado logrando ingresar al sistema y visualizar la página principal. Una vez ingresado en el sistema se seleccionó la opción de cerrar sesión y el resultado fue satisfactorio cerrando la sesión y devolviendo al usuario a la página de Login.</p>		

Fuente: Elaboración propia.

Tabla 64. *Caso de prueba de CRUD de entidad de usuario.*

Caso de prueba de CRUD de entidad de usuario	<Código del CP>	CP-02
<p>Descripción: En el siguiente caso de prueba se busca realizar el proceso completo de ingreso de un registro de un usuario, consulta del usuario ingresado, edición del usuario y por último el borrado del usuario.</p>		

<p>Prerrequisitos</p> <ul style="list-style-type: none"> - El usuario tiene que contar con credenciales previamente creados en el sistema. - El usuario tiene que conocer su nombre de usuario y contraseña. - El usuario debe de tener rol de administrador para realizar la operación. - El usuario tiene que contar con la información del usuario agregar.
<p>Pasos:</p> <p>Seguir los pasos enumerados en el caso de uso CU-03, Registro de usuarios. Seguir los pasos enumerados en el caso de uso CU-04, Consulta de usuarios. Seguir los pasos enumerados en el caso de uso CU-05, Modificación de usuarios. Seguir los pasos enumerados en el caso de uso CU-06, Eliminación de usuarios.</p>
<p>Resultado esperado:</p> <p>El sistema debe ser capaz de realizar un mantenimiento completo de un usuario contemplando los pasos de registro en el sistema, consulta, edición y borrado.</p>
<p>Resultado obtenido:</p> <p>En la prueba realizada se ejecutó el registro de un usuario colocando en el sistema los atributos de número de identificación, nombre, primer apellido, segundo apellido, número de teléfono, correo electrónico, contraseña y rol del usuario siendo estos usuario administrador o usuario final. El usuario luego de ingresar toda la información correspondiente y accionar el botón de ingresar, el usuario fue registrado en el sistema satisfactoriamente. Luego de esto se mostró dicho usuario en el listado de usuarios y se consultó la información detallada del usuario logrando visualizarla correctamente en pantalla. Se realizó la modificación del correo electrónico y el teléfono del usuario satisfactoriamente mostrando en pantalla los datos modificados. Y por último se borra el usuario obteniendo un resultado exitoso al no visualizarse en la lista de usuarios registrados.</p>

Fuente: Elaboración propia.

Tabla 65. Caso de prueba de CRUD de entidad de cliente.

Caso de prueba de CRUD de entidad de cliente	<Código del CP>	CP-03
<p>Descripción:</p> <p>En el siguiente caso de prueba se busca realizar el proceso completo de ingreso de un registro de un cliente, consulta del cliente ingresado, edición del usuario y por último el borrado del usuario.</p>		
<p>Prerrequisitos</p> <ul style="list-style-type: none"> - El usuario tiene que contar con credenciales previamente creados en el sistema. - El usuario tiene que conocer su nombre de usuario y contraseña. - El usuario debe de tener rol de administrador para realizar la operación. - El usuario tiene que contar con la información del cliente agregar. 		

<p>Pasos: Seguir los pasos enumerados en el caso de uso CU-07, Registrar cliente. Seguir los pasos enumerados en el caso de uso CU-08, Consultar cliente. Seguir los pasos enumerados en el caso de uso CU-09, Modificar cliente. Seguir los pasos enumerados en el caso de uso CU-10, Eliminar cliente.</p>
<p>Resultado esperado: El sistema debe ser capaz de realizar un mantenimiento completo de un cliente contemplando los pasos de registro en el sistema, consulta, edición y borrado.</p>
<p>Resultado obtenido: En la prueba realizada se ejecutó el registro de un cliente colocando en el sistema los atributos de tipo de identificación (física, jurídica, extranjería, NITE), número de identificación, nombre, primer apellido, segundo apellido, número de teléfono, correo electrónico, dirección de domicilio, provincia, cantón, distrito y disponibilidad de crédito. El usuario luego de ingresar toda la información correspondiente y presionar el botón de ingresar, el registro de cliente fue registrado en el sistema satisfactoriamente. Luego de esto se mostró dicho cliente en el listado de clientes y se consultó la información detallada del cliente logrando visualizarla correctamente en pantalla. Se realizó la modificación del correo electrónico y el teléfono del cliente satisfactoriamente mostrando en pantalla los datos modificados. Y por último se borra el cliente obteniendo un resultado exitoso al no visualizarse en la lista de clientes registrados.</p>

Fuente: Elaboración propia.

Tabla 66. Caso de prueba de CRUD de entidad de proveedor.

Caso de prueba de CRUD de entidad de proveedor	<Código del CP>	CP-04
<p>Descripción: En el siguiente caso de prueba se busca realizar el proceso completo de ingreso de un registro de un proveedor, consulta del proveedor ingresado, edición del proveedor y por último el borrado del proveedor.</p>		
<p>Prerrequisitos</p> <ul style="list-style-type: none"> - El usuario tiene que contar con credenciales previamente creados en el sistema. - El usuario tiene que conocer su nombre de usuario y contraseña. - El usuario debe de tener rol de administrador para realizar la operación. - El usuario tiene que contar con la información del proveedor a agregar. 		

<p>Pasos: Seguir los pasos enumerados en el caso de uso CU-12, Registrar proveedor. Seguir los pasos enumerados en el caso de uso CU-13, Consultar proveedor. Seguir los pasos enumerados en el caso de uso CU-14, Modificar proveedor. Seguir los pasos enumerados en el caso de uso CU-15, Eliminar proveedor.</p>
<p>Resultado esperado: El sistema debe ser capaz de realizar un mantenimiento completo de un proveedor contemplando los pasos de registro en el sistema, consulta, edición y borrado.</p>
<p>Resultado obtenido: En la prueba realizada se ejecutó el registro de un proveedor colocando en el sistema los atributos de número de cédula jurídica, nombre del proveedor, número de teléfono, correo electrónico, dirección exacta, provincia, cantón y distrito. El usuario luego de ingresar toda la información correspondiente y presionar el botón de ingresar, el registro de proveedor fue registrado en el sistema satisfactoriamente. Luego de esto se mostró dicho proveedor en el listado de proveedores y se consultó la información detallada del proveedor logrando visualizarla correctamente en pantalla. Se realizó la modificación del correo electrónico y el teléfono del proveedor satisfactoriamente mostrando en pantalla los datos modificados. Y por último se borra el proveedor obteniendo un resultado exitoso al no visualizarse en la lista de proveedores registrados.</p>

Fuente: Elaboración propia.

Tabla 67. *Caso de prueba de CRUD de entidad de producto.*

Caso de prueba de CRUD de entidad de producto	<Código del CP>	CP-05
<p>Descripción: En el siguiente caso de prueba se busca realizar el proceso completo de ingreso de un registro de un producto, consulta del producto ingresado, edición del producto y por último el borrado del producto.</p>		
<p>Prerrequisitos</p> <ul style="list-style-type: none"> - El usuario tiene que contar con credenciales previamente creados en el sistema. - El usuario tiene que conocer su nombre de usuario y contraseña. - El usuario debe de tener rol de administrador para realizar la operación. - El usuario tiene que contar con la información del producto a agregar. 		
<p>Pasos: Seguir los pasos enumerados en el caso de uso CU-17, Registrar producto. Seguir los pasos enumerados en el caso de uso CU-18, Consultar producto.</p>		

Seguir los pasos enumerados en el caso de uso CU-19, Modificar producto.
Seguir los pasos enumerados en el caso de uso CU-20, Eliminar producto.

Resultado esperado:

El sistema debe ser capaz de realizar un mantenimiento completo de un producto contemplando los pasos de registro en el sistema, consulta, edición y borrado.

Resultado obtenido:

En la prueba realizada se ejecutó el registro de un producto colocando en el sistema los atributos de código de producto, nombre de producto, descripción de producto, cantidad de producto, precio de venta, precio de costo, porcentaje de descuento y tipo de producto. El usuario luego de ingresar toda la información correspondiente y presionar el botón de ingresar, el registro de producto fue registrado en el sistema satisfactoriamente. Luego de esto se mostró dicho producto en el listado de producto y se consultó la información detallada del producto logrando visualizarla correctamente en pantalla. Se realizó la modificación del precio de venta y descuento satisfactoriamente mostrando en pantalla los datos modificados. Y por último se borra el producto obteniendo un resultado exitoso al no visualizarse en la lista de productos registrados.

Fuente: Elaboración propia.

Tabla 68. Caso de prueba de consulta y ejecución de pagos a clientes y proveedores.

Caso de prueba de consulta y ejecución de pagos a clientes y proveedores.	<Código del CP>	CP-06
<p>Descripción: En el siguiente caso de prueba, se ejecutan los procedimientos de consulta de los créditos para un determinado cliente y proveedor. A su vez se busca poder ejecutar satisfactoriamente el pago a un crédito de un cliente o de un proveedor.</p>		
<p>Prerrequisitos</p> <ul style="list-style-type: none"> - El usuario tiene que contar con credenciales previamente creados en el sistema. - El usuario tiene que conocer su nombre de usuario y contraseña. - Tiene que existir un cliente o proveedor previamente registrado en el sistema al realizar este caso de prueba. 		
<p>Pasos: Seguir los pasos enumerados en el caso de uso CU-11, Consulta de crédito de cliente. Seguir los pasos enumerados en el caso de uso CU-16, Consulta de crédito de proveedor.</p>		

Resultado esperado: El usuario debe ser capaz de poder visualizar la información crediticia para un determinado cliente o proveedor visualizando información importante como monto de la deuda, fecha de pago y monto de cuota. A su vez debe de poder registrar el pago de un crédito, de esta forma actualizando la información crediticia del cliente o proveedor.
Resultado obtenido: En el presente caso de prueba se realizó la consulta del crédito de un cliente y de un proveedor registrados previamente en el sistema, en donde se logra visualizar satisfactoriamente la información del crédito. De igual forma se consulta un cliente y un proveedor que no tenga deudas activas y el sistema mostro satisfactoriamente que no existía ninguna operación crediticia. Adicional al procedimiento de consulta se realizaron pagos a la operación de crédito para ambas entidades, siendo este un resultado exitoso en donde posterior a la prueba se actualizo la información crediticia.

Fuente: Elaboración propia.

Tabla 69. *Caso de prueba de cierre y apertura de caja.*

Caso de prueba de cierre y apertura de caja	<Código del CP>	CP-07
Descripción: En el siguiente caso de prueba se busca ejecutar una apertura de caja y su correspondiente cierre.		
Prerrequisitos <ul style="list-style-type: none"> - El usuario tiene que contar con credenciales previamente creados en el sistema. - El usuario tiene que conocer su nombre de usuario y contraseña. - Para el caso de una apertura no puede existir una apertura previamente realizada en el sistema. - Para el caso de un cierre tiene que realizar una apertura previamente creada en el sistema. 		
Pasos: Seguir los pasos enumerados en el caso de uso CU-23, Apertura de caja. Seguir los pasos enumerados en el caso de uso CU-24, Cierre de caja.		
Resultado esperado: Con la ejecución de este caso de prueba el usuario debe de ser capaz de realizar una apertura de caja para el inicio de operaciones de compra y venta, y a su vez realizar el cierre de caja al finalizar el día de operaciones.		

Resultado obtenido:

El usuario realizo una apertura de caja, ingresando el dinero con el que apertura la caja, al realizar la operación este género un registro de apertura con éxito y se le habilito la posibilidad al usuario de realizar operaciones de compra y venta satisfactoriamente. Luego de realizar algunos casos de prueba según el código de CP, CP-08 se procedió a realizar el cierre de ventas siendo este completado con éxito. El cierre de ventas logró visualizar información como totalización de ventas y compras, dinero inicial en caja, dinero final en caja, y el arqueo correspondiente mostrando algún posible faltante o sobrante de dinero.

Fuente: Elaboración propia.

Tabla 70. *Caso de prueba de compras y ventas.*

Caso de prueba de compras y ventas	<Código del CP>	CP-08
Descripción: En el siguiente caso de prueba se busca realizar con éxito una operación de venta y compra.		
Prerrequisitos <ul style="list-style-type: none"> - El usuario tiene que contar con credenciales previamente creados en el sistema. - El usuario tiene que conocer su nombre de usuario y contraseña. - Realizar previamente una apertura de caja según el código del CP, CP-08. - Contar con los productos a vender o comprar previamente registrados en el sistema. 		
Pasos: Seguir los pasos enumerados en el caso de uso CU-21, Registro de Venta. Seguir los pasos enumerados en el caso de uso CU-22, Registro de Compra.		
Resultado esperado: Al finalizar la ejecución de este caso de prueba se espera poder procesar con éxito una compra y venta, donde realizando este procedimiento se hagan las actualizaciones correspondientes de inventario y genere el comprobante respectivo de la operación.		
Resultado obtenido: Al ejecutar este caso de prueba el usuario realizo la compra de un producto previamente registrado en el sistema, se logra seleccionar un proveedor con éxito y el sistema muestra el listado de proveedores existentes en el sistema correctamente. Al finalizar la compra del producto se realizan con buenos resultados las operaciones de actualización de inventario agregando la cantidad de productos especificada en la compra y a su vez generando un comprobante con el detalle de la operación. Al finalizar la operación de compra se realizan		

pruebas creando una operación de venta, en donde el sistema fue capaz de visualizar el listado de clientes existentes en el sistema y a su vez la lista de productos disponibles para la venta. Al finalizar la operación de venta, se actualiza correctamente el inventario disminuyendo la cantidad de artículos según los artículos vendidos y adicional se genera el comprobante respectivo al finalizar la venta. Con esto se concluye que tanto las operaciones de venta como las de compra se realizaron con éxito.

Fuente: Elaboración propia.

Tabla 71. *Caso de prueba de análisis de datos.*

Caso de prueba de análisis de datos	<Código del CP>	CP-09
<p>Descripción: En el siguiente caso de prueba se espera poder visualizar los reportes de análisis de datos.</p>		
<p>Prerrequisitos</p> <ul style="list-style-type: none"> - El usuario tiene que contar con credenciales previamente creados en el sistema. - El usuario tiene que conocer su nombre de usuario y contraseña 		
<p>Pasos: Seguir los pasos enumerados en el caso de uso CU-25, Análisis de datos.</p>		
<p>Resultado esperado: Al finalizar el caso de prueba el usuario debe de ser capaz de visualizar los distintos reportes de análisis de datos e interactuar con ellos.</p>		
<p>Resultado obtenido: Una vez ejecutada esta prueba el usuario fue capaz satisfactoriamente de visualizar los distintos reportes de análisis de datos que ofrece el sistema.</p>		

Fuente: Elaboración propia.

Referencias

- Aguilar, C. y Anaya, M. (2015). *Desarrollo de un sistema de punto de venta dirigido a papelerías*. <https://pdfcoffee.com/sistema-de-punto-de-ventas-pdf-pdf-free.html>.
- Cervantes, J., Gómez, M. González P. y García, A. (2016). *Introducción a la programación orientada a objetos*. http://www.cua.uam.mx/pdfs/revistas_electronicas/libros-electronicos/2016/2intro-poo/programacion_web.pdf.
- Ferrero, P. (2015). *La gestión de inventarios, aplicación práctica en una empresa del sector farmacéutico, el caso de laboratorios Jiménez, SL*. https://buleria.unileon.es/bitstream/handle/10612/4559/45688755z_gade_julio15%20pdf.pdf?sequence=1.
- Hernández Sampieri, R. (2017). *Fundamentos de Investigación*. McGraw-Hill Interamericana de España S.L.
- Herrera, J. (2008). *La investigación cualitativa*. <https://juanherrera.files.wordpress.com/2008/05/investigacion-cualitativa.pdf>
- Instituto de Normas Técnicas de Costa Rica [INTECO]. (2009). *Ingeniería del Software: Metodologías y ciclos de vida*. academia.edu/22159265/INGENIERÍA_DEL_SOFTWARE_METODOLOGÍAS_Y_CICLOS_DE_VIDA_Laboratorio_Nacional_de_Calidad_del_Software?from=cover_page.
- Maranto y González, M. (2015). *Fuentes de Información*. <https://repository.uaeh.edu.mx/bitstream/bitstream/handle/123456789/16700/LECT132.pdf>
- Mateu, C. (2004). *Desarrollo de aplicaciones web*. <https://libros.metabiblioteca.org/bitstream/001/591/1/004%20Desarrollo%20de%20aplicaciones%20web.pdf>.
- Ministerio de Trabajo y Seguridad Social [MTSS]. (2021). *Lista de Salarios Mínimos del Sector Privado*. mtss.go.cr/temas-laborales/salarios/Documentos-Salarios/lista_salarios_2022.pdf.

- Navarro, E. y Ortega, L. (2014). *Desarrollo de un software de facturación parametrizable para empresas de venta directa por catálogo*.
<https://bibdigital.epn.edu.ec/bitstream/15000/7415/1/CD-5578.pdf>.
- Ozz, E. (2006). *Administración de los sistemas de información* (5ª edición).
<http://cotana.informatica.edu.bo/downloads/Administracion%20SI.pdf>.
- Pérez, D. y Pérez, I. (2006). *El Producto: Concepto y Desarrollo*.
<https://static.eoi.es/savia/documents/componente45111.pdf>.
- Pressman, R. (2005) *Un Enfoque Práctico*. McGraw-Hill.
<http://cotana.informatica.edu.bo/downloads/IdIngenieria.de.software.enfoque.practico.7ed.Pressman.PDF>
- Ricardo, C. (2006). *Bases de Datos*. <http://cotana.informatica.edu.bo/downloads/Id-Bases%20de%20datos.pdf>.
- Salazar, E. (2012). *Prototipo, el contexto, y la ingeniería del software*.
academia.edu/11217243/Prototipo_Contexto_e_Ingeniería_del_Software.
- Sangri, A. (2014). *Administración de compras, adquisiciones y abastecimiento*.
<https://editorialpatria.com.mx/mobile/pdf/files/9786074386202.pdf>.
- Santesmases. (2009). *Fundamentos de Marketing*. Pirámide, Madrid. Recuperado de:
<https://repositori.uji.es/xmlui/bitstream/handle/10234/49394/s74.pdf>
- Serrano, A. (2010). *Proceso de ventas y herramientas estratégicas para la toma de decisiones para la banca personal del grupo HELM*.
<https://intellectum.unisabana.edu.co/bitstream/handle/10818/6900/125397.pdf;jsessionid=74EC60CE03F4582932B3EE21FF444AE6?sequence=1>.
- Vega Sánchez, D. G. (s.f.). *Gestión estratégica del Depto de Ventas Aplicada en una empresa Comercial-Farmacéutica*. <https://infolibros.org/pdfview/2435-aspectos-teoricos-marketing-y-ventas-dolly-giannina-vega/>.
- Wellness & Sport Consulting. (2021). *Manual de atención al cliente y proceso de ventas*.
<https://docer.com.ar/doc/ses0xxn>.

APENDICES

Apéndice A. Guía de Entrevista Semiestructurada Aplicada.

Guía de Entrevista Semiestructurada

Nombre de Entrevistador	Diego Mendez
Nombre de Entrevistado	Roxana Barboza
Fecha	Miércoles 28 de diciembre

- 1) ¿Cuál es su experiencia utilizando sistemas de computación, y que aplicaciones está familiarizado utilizar?

Actualmente en el negocio contamos con el servicio de impresión de documentos y fotocopiado, por esta razón utilizamos con bastante frecuencia las computadoras en aplicaciones como Word o Power Point. Mas de esas aplicaciones con estamos familiarizados.

- 2) ¿Anteriormente ha utilizado un sistema de facturación, y está familiarizado con el concepto?, si la respuesta es positiva hable un poco de la experiencia con el sistema

No he escuchado mucho del tema, pero entiendo que es lo que utilizan en los supermercados al momento de pagar en la caja. Anteriormente nunca hemos utilizado este tipo de sistemas.

- 3) ¿En caso de no contar con un sistema de información, el negocio posee algún otro tipo de sistema tal como caja registradora?

Si, hace años que compramos una caja registradora y es la que utilizamos actualmente no siempre es necesario entregar el ticket de caja al cliente, pero en caso de que el cliente lo requiera le entregamos el ticket de caja que entrega la misma caja registradora, también la caja registradora nos ayuda a saber cuánto hemos vendido al final del día.

- 4) ¿Podría indicarme como es el proceso actual al momento de hacer una venta, en donde se contabiliza y si actualmente se tiene un control de entrada o salida de dinero?

Cuando hacemos una venta sumamos todos los productos en la caja registradora, al final en la caja registradora ponemos con cuanto paga el cliente y nos dice cuanto cambio tenemos que entregar. Adicional a esto tenemos un cuaderno donde vamos apuntando todas las ventas que se realizan durante al día, cuando terminamos el día generamos el cierre en la caja registradora para observar el total de ventas, posterior a esto contamos el dinero, y comparamos también con lo que tenemos apuntado en el cuaderno.

- 5) ¿Podría mencionarme el proceso actual al momento de comprar productos a los proveedores, como se registran las facturas o se llevan control de las cuentas por pagar en caso de compras a crédito?

A como lo hacemos ahorita, únicamente archivamos las facturas para tener la documentación de lo que se ha comprado a mano, y en caso de que sean compras que hacemos a crédito tenemos un control en un cuaderno donde tenemos todas las deudas por pagar. Nos apoyamos también con un calendario para tener un control de las fechas.

- 6) ¿Cuál es el proceso actual para el control de directorio de clientes, en aspectos tales como teléfonos, correos electrónicos, o cualquier información de contacto relevante que se requiera documentar?

En este momento no contamos con ningún procedimiento para registrar información de los clientes en algunos casos con clientes muy frecuentes tenemos una libreta con números de teléfono y nombres, pero en este momento es con lo único que contamos.

- 7) ¿El negocio actualmente realiza ventas a crédito? En caso de ser positiva la respuesta como se maneja el control de fechas por cobrar al cliente y en forma general como se maneja la línea de crédito.

Si contamos con el servicio de compras a créditos, pero únicamente para ciertos clientes no extendemos esta posibilidad a todas las personas ya que sería muy complicado para nosotros tener el control de todas estas cuentas. Como estamos ubicados cerca de colegios a las juntas directivas de los colegios si les damos crédito en servicios de impresión y cosas así, también en algunos clientes de confianza tenemos este servicio de compras a crédito. Con respecto al control de compras de forma muy similar con los proveedores tenemos un cuaderno donde registramos estas compras y nos apoyamos con un calendario para recordar las fechas de pago.

- 8) ¿Podría indicarme cómo se realiza actualmente el manejo de inventarios, en aspectos como control de existencias, precios de producto, características que sean relevantes recopilar acerca de los productos?

El control de inventarios lo realizamos únicamente una vez por año para el cierre fiscal para este momento hacemos un conteo total de todos los productos que esto lo presentamos al contador. Y no recopilamos la información de los productos como precios o similar sabemos los precios porque los conocemos o bien porque marcamos el precio en todos los productos. Cuando se acaba completamente el producto, en ese punto es donde compramos más al proveedor.

- 9) ¿Cuál comprobante al momento de la venta se le entrega al cliente tal como algún tipo de factura membretada en específico, factura digital, o bien solamente tiquete de caja?

Lo que se le entrega al cliente es el tiquete de caja que genera la caja registradora.

10) ¿Cuál régimen tributario rige en la librería?

Actualmente estamos con el régimen simplificado.

11) ¿Cuántos empleados operan actualmente en la librería?

Normalmente somos 3 personas atendiendo el local, sin embargo, en épocas altas como matriculas e inicios de lecciones conseguimos 2 personas más que nos ayuden en el negocio.

12) ¿En caso de ser varios empleados los que operan la librería, todos tienen las mismas autoridades dentro del negocio o existe algún nivel jerárquico al momento de las distintas operaciones?

Únicamente yo y mi esposo nos encargamos de temas como compras de productos, manejo de proveedores, el tema de compras a créditos que le comenté anteriormente, también el tema de conteo de dinero al final de día lo hacemos nosotros mismos. La caja la utilizamos todos por lo tanto no hay puesto de cajero como tal. En caso de decir un nivel jerárquico yo y mi esposo seríamos los administradores.

13) ¿Como se maneja los cierres de caja al final de día?

Esto lo generamos desde una opción en la caja registradora que nos emite un ticket con las compras que se realizaron durante el día.

14) En un término operacional, ¿Que se espera obtener con el prototipo funcional (Que módulos o que funcionalidades espera que cumpla el prototipo)?

Realmente como le comenté anteriormente no conozco mucho de este tipo de sistemas, pero si tendría que decir de lo que espero, es que cumpla con las mismas funciones que actualmente tenemos con la caja registradora, y que nos ayude con

temas de inventarios ya que quisiéramos tener más control de los productos en cuestión de existencias y precios de los productos, también sería de mucha ayuda si el sistema maneja las compras y ventas a crédito para el control de cuentas por pagar y cuentas por cobrar. Un tema también muy interesante sería el control de información de clientes que comento en una de las preguntas anteriores.

- 15) ¿Qué información espera obtener en los reportes arrojados por el prototipo, tales como reportes de clientes, inventarios o ventas por decir algunos ejemplos?

Nos gustaría que el sistema tenga reportes como la lista de productos, la lista de clientes, la lista de ventas realizadas durante un día al igual que el listado de compras realizadas.

- 16) ¿Qué información desea obtener del prototipo en la factura o ticket generado al cliente?

Tendría que ser la misma estructura que tiene actualmente el ticket que genera la caja registradora, la información de la librería, nombre, teléfono, dirección, numero cedula, y un detalle de los productos con la cantidad y su respectivo precio. Sería de mucha utilidad que tenga el nombre de la persona que realiza la venta esto nos sería de mucha utilidad para tener más control.

- 17) En términos generales, ¿Qué características espera obtener en la interfaz gráfica del prototipo?

Lo que espero realmente es que el sistema sea fácil de utilizar ya que no estamos muy familiarizados a este tipo de aplicaciones, que no haya que completar muchas cosas también ya que cuando hay mucho movimiento en la librería sería complicado estar llenando más cosas y hacer más lento el trabajo.

Apéndice B. Guía de Observación Aplicada.

Guía de Observación

Fecha	Miércoles 28 de diciembre 2022
--------------	--------------------------------

Apartado I - Hardware

Tabla 72, Apéndice B. *Observaciones de hardware.*

Aspecto	Observaciones
Características de computadoras.	<p>El negocio cuenta con dos computadoras una con mayor recurso que la otra a continuación se detallan ambas:</p> <p>Computadora 1</p> <p>Marca: HP Procesador: CoreI3 Memoria: 4 GB Disco Duro: HDD 500 GB</p> <p>Computadora 2</p> <p>Marca: Dell Procesador: CoreI8 Memoria: 16 GB Disco Duro: SDD 500 GB</p>
Dispositivos periféricos presentes en la librería (Monitores, teclados, impresoras, etc.)	<p>El negocio cuenta con monitores para ambas computadoras, teclados y mouses. De igual forma, tienen varias impresoras ya que una de las líneas de negocio es el servicio de impresión a su vez también varias fotocopiadoras que también son utilizadas para imprimir y escanear desde las computadoras.</p>

Características del equipo para alojar el sistema y la base de datos	Según las características observadas, el sistema se podría alojar en la computadora 2 que presenta las características mencionadas anteriormente.
Redes presentes en los equipos.	Los equipos están interconectados por medio de un router inalámbrico con conexión a internet de 100mb de velocidad de subida y 20 mb de velocidad de bajada.

Fuente: Elaboración propia.

Otras Observaciones: El negocio no cuenta con una impresora de tiquetes para el sistema de facturación por lo tanto es una compra que se recomienda realizar para la utilización del prototipo.

Apartado II - Software

Tabla 73, Apéndice B. *Observaciones de software.*

Aspecto	Observaciones
Sistemas Operativos utilizados en los equipos de la librería. Tipo y Versiones.	Ambas computadoras cuentan con Windows 11 instalado.

Sistemas de ventas presente o dispositivos alternativos tales como cajas registradoras.	El negocio cuenta con caja registradora.
---	--

Fuente: Elaboración propia.

Otras Observaciones:

Apartado III - Empleados

Tabla 74, Apéndice B. *Observaciones de empleados.*

Aspecto	Observaciones
Cantidad de empleados que operan en la librería.	Al momento de la visita de observación se encontraban únicamente 2 personas trabajando, sin embargo, el administrador indica que en época alta normalmente trabajan 5 personas en la librería.
Puestos o funciones identificadas en los distintos empleados.	No hay puestos bien definidos sin embargo hay 2 administradores y los demás son colaboradores.

Tareas frecuentes de los Empleados	Los colaboradores realizan tareas como servicios de impresión, fotocopiado, escaneado de documentos y ventas de productos. Los administradores son los únicos que se encargan de temas de inventario, proveedores y compras a su vez el cierre diario es operado por los administradores del negocio.
Nivel de uso de los empleados en computadoras (Nulo, Básico, Intermedio o Avanzado)	El nivel de uso en computación de los colaboradores es básico en algunos casos e intermedio en otros.

Fuente: Elaboración propia.

Otras Observaciones: Todos los empleados realizan el procedimiento de registrar la venta en la caja registradora como tal no existe un puesto definido de cajero, por lo tanto, el colaborador que realiza la venta el mismo ingresa la venta en la caja registradora.

Apartado IV - Ventas

Tabla 75, Apéndice B. Observaciones de Ventas.

Aspecto	Observaciones
Proceso de registro de ventas (Histórico de Ventas)	Cuando se realiza una venta en el negocio esta se ingresa en la caja registradora, la caja se encarga de sumar los productos e indicarle al colaborador cuanto cambio tiene que entregar al cliente. Adicional en un cuaderno se va anotando las ventas que se van realizando a lo largo del día.

Proceso de registro de productos en una venta.	Los productos se registran por medio de códigos en la caja registradora, esta se encarga de sumarlos al momento de la venta. No hay un proceso adicional como un registro de inventario en donde se indique cuando hay salida de productos.
Proceso de aplicación de impuestos o descuentos en una venta y totalización de venta.	El precio que se le coloca a los productos ya es con el impuesto del 13% aplicado, de igual forma si se aplica un descuento en algún producto está contemplado en el precio final del producto. Con respecto a la totalización de la venta esta se encarga la caja registradora y a su vez le indica al usuario cuanto cambio tiene que darle al cliente.
Tiquete o factura generado al cliente.	El tiquete de caja generado por la caja registradora es lo que se le entrega al cliente.
Métodos de pago en la librería.	El negocio acepta únicamente efectivo y SINPE móvil actualmente no cuenta con datafono para pagos con tarjeta.

Tipos de ventas (Contado o Crédito)	El negocio cuenta con ambos tipos de venta, las ventas de contado son las más comunes ya que el negocio únicamente ofrece ventas a crédito a ciertos cliente que ya los administradores determinaron previamente, no se puede otorgar ventas a crédito sin autorización de los administradores del negocio.
Proceso de control de ventas a crédito.	El control de ventas a crédito se maneja en un cuaderno, cuando se vende a crédito esta se apunta y en este mismo cuaderno se registran los pagos que realiza el cliente a la cuenta, se le entrega un recibo cuando se realiza un pago a la cuenta. Para recordar las fechas de pago los administradores del negocio se apoyan con un calendario.
Registro de información de contacto de clientes.	Actualmente el negocio no cuenta con un proceso definido para esto, únicamente tienen un pequeño de directorio de clientes y proveedores, pero realmente no es a todos los clientes que se les registra información.

Fuente: Elaboración propia.

Otras Observaciones: Se observa que en el proceso de ventas no existe un proceso paralelo en el que se pueda actualizar el inventario al momento de la salida de productos.

Tabla 76, Apéndice B. Observaciones de compras.

Aspecto	Observaciones
Proceso de registro de compras (Histórico de Compras)	En este momento únicamente los administradores tienen privilegios de realizar compras en el negocio, cuando se realiza una compra el único proceso que se tiene es el archivado de la factura para efectos de control del histórico de compras realizadas.
Proceso de control de compras a crédito	Para el proceso de control de compras a crédito se maneja de una forma muy similar que con los clientes en donde se tiene un cuaderno donde se registran estas compras a crédito, y aquí mismo se registran los pagos que realizan los clientes. Para el manejo de fechas por pagar se apoyan con un calendario en donde registran los pagos que se tienen que realizar en de terminado día.
Proceso de registro de inventario al realizar compras.	No hay un proceso de inventario como tal al momento de una compra, cuando se compran productos el procedimiento es calcular los impuestos correspondientes, descuentos en caso de aplicarse, y colocarle una etiqueta al producto con el precio final para el consumidor.
Proceso de registro de información de contacto de proveedores.	Actualmente el negocio no cuenta con un proceso definido para esto, únicamente tienen un pequeño de directorio de clientes y proveedores, pero realmente no es a todos los clientes que se les registra información.

Fuente: Elaboración propia.

Otras Observaciones:

Apartado VI - Inventario

Tabla 77, Apéndice B. Observaciones de Inventario.

Aspecto	Observaciones
Proceso de registro de inventario (Entradas y Salidas de productos)	El negocio no cuenta con un procedimiento en el cual se alimente el inventario cuando se compran o venden productos (Entradas y salidas) para llevar un control del inventario este se realiza una vez por año al momento del cierre fiscal.
Proceso de registro de información de producto (Existencias, precio, descripción, etc.)	Con respecto a los productos el único atributo que maneja el negocio es el precio el cual es calculado al momento de la compra del producto en donde se calculan los impuestos respectivos, la ganancia, y descuentos en caso de aplicarse. Adicional a esto no se captura otras características del producto.
Manejo de agotamiento de existencias.	No existe un procedimiento que notifique al negocio cuando se está cerca de agotar las existencias de un determinado producto. Cuando se realiza la compra del producto es cuando ya no hay disponibles para la venta.

Fuente: Elaboración propia.

Otras Observaciones:

Apartado VII – Cierres de Caja

Tabla 78, Apéndice B. *Observaciones de cierre de caja.*

Aspecto	Observaciones
Proceso de Cierre de Caja	El cierre de ventas diarios se realiza por medio de la caja registradora, de igual forma se apoya con el cuaderno donde diariamente se apuntan todas las ventas que se realizan, se hace un control cruzado entre ambos elementos más el conteo de dinero que haya en caja al final de día. En caso de haya una incongruencia en estos valores se revisa en donde puede estar el fallo.

Fuente: Elaboración propia.

Otras Observaciones: